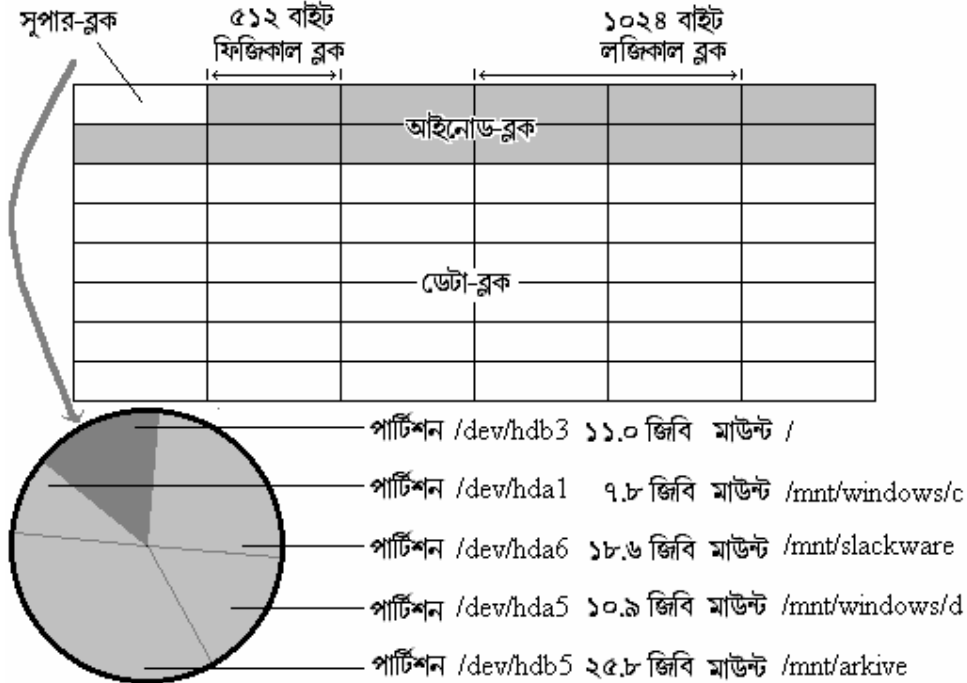
গ্নু-লিনাক্স ইশকুল

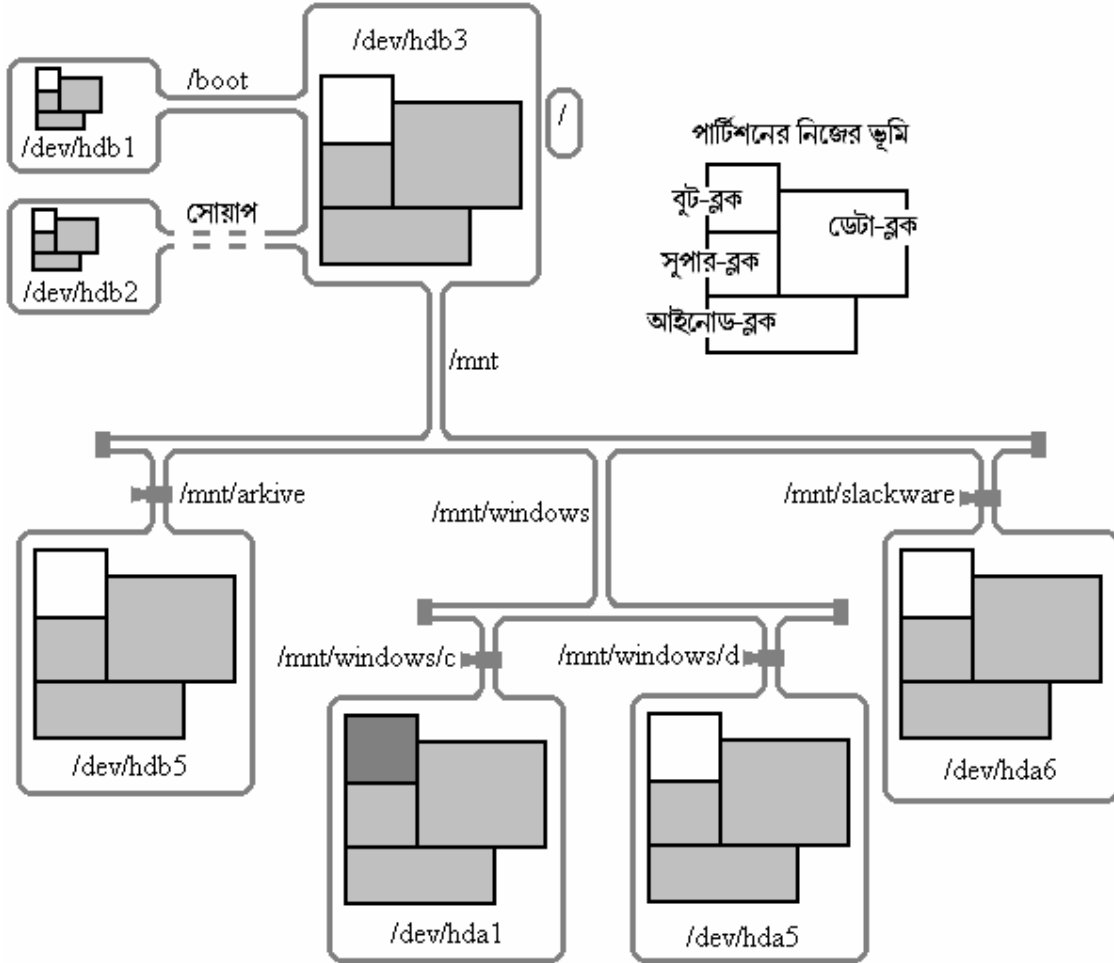
জিএলটি মধ্যমগ্রাম
glt-mad@ilug-cal.org

আজ পাঁচই জানুয়ারি, এখন বেলা বারোটাতেও হাড় কেঁপে যাচ্ছে, আরো আমার লেখার টেবিলের পাশেই দুদুটো জানলা দিয়ে উত্তর দিকের পুকুরের হাওয়া। তবে শীতই একমাত্র কারণ নয় ফের লেখায় ফেরার দেরির, এর মধ্যে এই সিরিজেরই তিন নম্বর দিন মানে চার নম্বর চ্যাপ্টারটাকে অপর বইমেলা সংখ্যার জন্যে আলাদা একটা প্রবন্ধ হিশেবে ছাপতে দিলাম, আর আলাদা করে নেওয়া মানেই কিছু টুকরোটাকরা মেরামত করতেই হয়। আর ওই করতে গিয়ে শ্যাননের 'এ ম্যাথামেটিকাল থিওরি অফ কমিউনিকেশনস' নেটে পেয়ে গেলাম, সেটা পড়বার লোভটা সামলাতে পারছিলাম না। যাকগে আবার আমি ফেরত এসেছি আমাদের গ্নু-লিনাক্স গোষ্ঠীর পিণ্ডদানে, যাকে ভদ্রলোকের ভাষায় বলে 'পেয়িং ব্যাক টু দি কমিউনিটি' — সেই ইয়ং বেঙ্গল আর বেম্মো মিশনারিদের সময় থেকে ভদ্রলোকরা তো ইংরিজিতে ছাড়া বোঝেনা।

।।দিন আট।।

১।। ডিরেক্টরি আর কারনেল

সাত নম্বর দিনে যে আলোচনাটা হয়েছিল, সেটা খুব বড় হয়ে যাচ্ছিল বলে মাঝপথে দুম করে থামিয়ে দিয়েছিলাম, একটু মনে পড়িয়ে দিই। আমরা একটা গ্নু-লিনাক্স সামগ্রিক ব্যবস্থা বা ঐক্যবদ্ধ ফাইলসিস্টেমের এক একটা অংশ হিশেবে এক একটা পার্টিশনের এক একটা আলাদা ফাইলসিস্টেমের খুঁটিনাটি আলোচনা শুরু করেছিলাম, যে পার্টিশনগুলোর এক একটা আলাদা আলাদা ডিরেক্টরিতে মাউন্ট করি আমরা। — শেষ বাক্যটায় দুবার 'ফাইলসিস্টেম' কথাটা দুটো অর্থে ব্যবহৃত — মনে পড়ছে? সেটা করতে গিয়ে আমরা বিভিন্ন ধরনের ব্লকের কথা বলছিলাম, আর একটু মনে পড়িয়ে দিই, একটা ছবি দিয়ে।

এই ছবিতে দেখুন, আমরা শুধু তিন রকমের ব্লক দেখিয়েছি। সুপার-ব্লক, আইনোড-ব্লক আর ডেটা-ব্লক। এক রকম ব্লক বাদ দিয়েছি। কোন রকম ব্লক বলুন তো? এটা এখুনি মনে করতে পারবেন। বুট-ব্লক দেখাইনি। কিন্তু এর পরের প্রশ্নটা এর চেয়ে আর একটু কঠিন। কেন? কেন দেখাইনি আমরা? নিচে আমরা মোটামুটি সাইজ অনুযায়ী একটা পিঠের পাঁচ টুকরোয় পাঁচটা পার্টিশনকে দেখিয়েছি। তার প্রথমটার ব্লক-গুলোর ছক এখানে আঁকা। এটা হল '/dev/hdb3', যেখানে

মূল সুজে সিস্টেমটা থাকে, যার ডিরেক্টরি '/', এই পার্টিশনে কোনো বুট ব্লক থাকবে না, মানে বুট-ব্লক ফাঁকা থাকবে। এই পাঁচটা পার্টিশনের একটাতেই মাত্র বুট-ব্লক থাকবে। কেন, মনে করতে পারছেন? ভাবুন একটু। পরের মাউন্ট নিয়ে ছবিটা একবার দেখুন। এই ছবিটা এখন পুরোটা বুঝতে পারবেন না, এই ছবিটার কথায় আমরা আবার আসব। উপরে দেখুন আমরা ফিজিকাল ব্লক আর লজিকাল ব্লক দুটোকেই দেখিয়েছি। এখানে পাঁচটা পার্টিশনের পাঁচটা মাউন্টপয়েন্ট দেখিয়েছি, পাঁচটা ডিরেক্টরি। মাউন্ট ব্যাপারটার কেজো ছকে আপনারা ইতিমধ্যেই পরিচিত হয়েছেন, কিন্তু মাউন্ট-এর গোটা ব্যাপারটা এখনো আমাদের আলোচনায় আসেনি। একটু বাদেই আসব। আর একটা জিনিষ মনে রাখবেন, ছয় এবং সাত নম্বর দিনে দেওয়া আমার মেশিনের পার্টিশন টেবিলের তালিকার সঙ্গে মেলান, দেখবেন, আমরা দুটো পার্টিশনকে বাদ দিয়েছি, সোয়াপ পার্টিশন মানে '/dev/hdb2', আর সুজে সিস্টেমের বুট ফাইল যেখানে থাকে, সেই বুট ডিরেক্টরিতে যে পার্টিশন মাউন্ট করা হয়, মানে '/dev/hdb1'। এরা অন্যগুলোর তুলনায় বড্ড ছোট বলে। আর খেয়াল রাখুন, নিচে যেটা আমরা একটা পিঠের টুকরো করে দেখিয়েছি, আর একটু বাস্তব করতে চাইলে সেটাকে দুটো পিঠের টুকরোয় দেখানো উচিত, কারণ, হার্ডডিস্ক দুটো। পার্টিশনের নামগুলো থেকে খেয়াল করুন। আর, এখানে পিঠেটাকে আমার মেশিনের মোট ডিস্কভূমি বলে ভাবুন। যার মধ্যে বাস্তবে দুটো হার্ডডিস্ক আছে।

এবার আসুন আমরা একটা ডিরেক্টরিকে বোঝার চেষ্টা করি। ডিরেক্টরি ফাইলের অপৌরুষেয় বিদঘুটেপনা নিয়ে আমরা আগেই বলেছি, কারনেল ছাড়া কারুর পড়ার অধিকার নেই ডিরেক্টরি ফাইল, মায় রুটেরও না, আর সবাই শুধু দেখতে পারে সেই ডিরেক্টরিতে কী আছে, তাও যদি তার সেই ডিরেক্টরি দেখার অনুমতি থাকে, আর রুটের তো সব ডিরেক্টরি দেখারই অনুমতি থাকে। একটা ডিরেক্টরিতে একটা ফাইলের দুটো ব্যাপার লিপিবদ্ধ থাকে, এক, ফাইলটার নাম। আর দুই, তার আইনোড নম্বর। কেন আইনোড ব্লকগুলোর তালিকায় ফাইলের নামটা থাকেনা, সাত নম্বর দিনের

আলোচনায় আমরা বলেছি, এবারে বুঝতে পারছেন? ডিরেক্টরির কাছে যেহেতু একটা ফাইলের নাম আর আইনোড নম্বর দুই-ই লেখা থাকে, সে এবার সহজেই ওই আইনোড নম্বর দিয়ে ফাইলটাকে পেয়ে যেতে পারে। ডিরেক্টরি ফাইল আর আইনোড-ব্লক এই দুইয়ে মিলে একটা পরস্পর-সম্পর্কিত যৌথ-তালিকা তৈরি করে। আইনোড নম্বর হল এই দুটো তালিকার মধ্যে সম্পর্কটা। একটা ফাইলের সমস্ত কিছু যদি আমরা জানতে চাই — খুব সহজ, ফাইলটার আইনোড নম্বর দিয়ে দুটো তালিকারই ওই বিশেষ নম্বর লাইনটা পড়ে ফেলো।

সাত নম্বর দিনের আলোচনায় আমরা কী দেখলাম — যখনই একটা ফাইলের একটা লিংক বানানো হচ্ছে, তার আইনোডে উল্লিখিত লিংকের সংখ্যাটা এক করে বেড়ে যাচ্ছে। আর এই লিংক ফাইলটার নতুন ফাইলনামের জন্যে ডিরেক্টরিতেও একটা নতুন এন্ট্রি তৈরি হচ্ছে। যার আইনোড নম্বরটা নিট কপি করে দেওয়া হচ্ছে মূল ফাইলটার থেকে, যার সে লিংক। আপনার বানানো লিংকগুলোর যে কোনো একটাকে এবার '`rm`' করে দেখুন, লিংকটা উড়ে যাবে, আর তার সঙ্গে সঙ্গে '`ls -il`' করে বানানো আইনোড-সহ দীর্ঘ তালিকায় মূল ফাইলটার লিংক-সংখ্যা এক কমে গেল। এখন আমরা জানি এর মানে কী — আইনোড-ব্লকে লেখা লিংক-সংখ্যাকে এক কমিয়ে দেওয়া হয়েছে। আর ডিরেক্টরি ফাইলে এই লিংকটার জন্যে বানানো এন্ট্রিটাও উড়িয়ে দেওয়া হয়েছে। একটা ফাইলের শেষ লিংকটা উড়িয়ে দেওয়া হলে, তখন ফাইলটাই মুছে গেল। তার ডেটা-ব্লকগুলো ধুয়ে মুছে পোস্কার করে রাখা হল, নতুন ফাইল লেখার জন্যে। আপনার সঙ্গে এই পৃথিবীর শেষতম লিংক তো আপনি নিজেই, এটা কখনো খেয়াল করেছেন? আমাদের মতো যাদের ছদ্মনামে লেখার অভ্যেস তৈরি হয়, তাদের কাছে এটা খুব স্পষ্ট। ত্রিদিব সেনগুপ্তর লেখা সামনে এনে ধরে দিতে পারে কেবল একমাত্র দীপঙ্কর দাশই, ত্রিদিব সেনগুপ্ত হল দীপঙ্কর দাশের লিংক ফাইল। ত্রিদিব সেনগুপ্ত মুছে গেলেও দীপঙ্কর দাশ থাকবে। কিন্তু দীপঙ্কর দাশ নিজে দুনিয়ার ডিস্ক থেকে মুছে গেলে, আর সে নিজেই নিজের লেখাকে হাজির করতে পারবে না। তার জায়গায় অন্য ফাইল এলো, এখন থেকে অন্য কেউ অপরের লেখা দিতে দেরি করায় সোমনাথের কাছে খিস্তি খাবে।

এবার, কারনেলের কাছে, সিস্টেমের কাছে, একটা ফাইলের সমস্ত হালহকিকত, সমস্ত ঠিকানা কী ভাবে থাকে, সেটা আমাদের কাছে স্পষ্ট। এবার ধরুন আপনি আপনার তৈরি 'onefile' ফাইলটায় কী আছে দেখতে চাইলেন। আপনি তার কমান্ড জানেন, দিলেন কষে '`cat onefile`'। সামনে তো স্ক্রিন জুড়ে গোটা গোটা অক্ষরে 'onefile' ভেসে উঠল। কিন্তু স্ক্রিনকা পিছে কেয়া হ্যায়? কারনেল এই যে 'onefile' ফাইলটাকে খুঁজে পেতে সামনে এনে আপনার স্ক্রিনে মানে স্ট্যান্ডার্ড আউটপুটে ধরে দিল, তার মানে আগে তাকে 'onefile' ফাইলটার ঠিকানা খুঁজে পেতে হল, সেখান থেকে ফাইলটার শুরুর ব্লকের ঠিকানা আর শেষ ব্লকের ঠিকানা পেতে হল, শেষ অব্দি পড়ে নিতে হল, কারণ, এক স্ক্রিনে যদি শেষ অব্দি না-ধরে তাকে পরের স্ক্রিনে পরের স্ক্রিনে পরের স্ক্রিনে ... দেখিয়ে যেতেই হবে। এর জন্যে স্টেপ বাই স্টেপ তার কাজগুলোকে ভাবুন।

এক, আগেই আমরা বলেছি কারনেল কী ভাবে কারেন্ট ডিরেক্টরির, মানে এখন যেখানে আছেন, কাজ করছেন, সেই ডিরেক্টরির ডিরেক্টরি ফাইলের আইনোড নম্বরকে মেমরিতে তুলে নেয়। যদি মনে না পড়ে সাত নম্বর দিনের ১১ নম্বর সেকশনের সাবসেকশনগুলো একবার দেখে আসুন। এই আইনোড নম্বরটাকে কারনেল এবার খুঁজে বার করে আইনোড-ব্লকগুলো থেকে। সেখান থেকে এই ডিরেক্টরি ফাইলের আইনোডটা বার করে। দুই, এই ডিরেক্টরির আইনোড থেকে এবার কারনেল বার করে এই ডিরেক্টরি ফাইলের ডেটা-ব্লকের ঠিকানা। তিন, ডিরেক্টরি ফাইলের ডেটা-ব্লক পড়ে কারনেল এবার খুঁজে পায় 'onefile' ফাইলটাকে এবং পেয়ে যায় 'onefile' ফাইলটার আইনোড নম্বর। চার, 'onefile' ফাইলের আইনোড নম্বর পেয়ে যাওয়া মাত্রই কারনেল আবার ফেরত যায় আইনোড-ব্লকে। সেখানে গিয়ে 'onefile' ফাইলের আইনোড খুঁজে বার করে। পাঁচ, এই আইনোড থেকে কারনেল এবার পড়ে নেয় 'onefile' ফাইলটার আয়তন, এবং ডিস্কের শরীরে তার ডেটা-ব্লকের ঠিকানাগুলো। এখানে ইনডিরেক্ট ব্লক যদি থাকে তাহলে সেইটাও পড়ে বাড়তি ডেটা-ব্লকের ঠিকানাগুলোও পেয়ে যায়। ছয়, এবার কারনেল যায় তার ডিস্ক ড্রাইভারের কাছে। তাকে গিয়ে বলে তোমার ডিস্কের হেডগুলোকে নড়িয়ে তুমি তোমার প্ল্যাটারের গায়ে এত নম্বর সিলিন্ডারের এত এত সেক্টর থেকে এই এই ব্লকগুলোকে পড়ে ফেলো। এই হল ফাইলটার আয়তন, এই বাইটসাইজ হওয়া অব্দি তুমি পড়ে চলো। এইটা মিলে গেলে কারনেল বোঝে যে তার গোটা ফাইলটা পড়া হয়েছে, এবং তখন তার পড়ে ফেলা তথ্যটা ফুটিয়ে তোলার পালা, তার জন্যে সে যাবে ডিসপ্লে ড্রাইভারের কাছে। সে অন্য গল্প।

২।। সোয়াপ পার্টিশনের স্বতন্ত্র ফাইলব্যবস্থা

এতবার আলাদা আলাদা ভাবে দুটো অর্থে ব্যবহার করা হয়েছে যে আমরা এখন 'ফাইলসিস্টেম' শব্দটার এই দুটো ব্যবহারের সঙ্গে অভ্যস্ত। আমরা যখন '/' থেকে শুরু করে গোটা ডিরেক্টরিব্যবস্থাটার কথা ভাবছি, সেটা একটা ঐক্যবদ্ধ ফাইলব্যবস্থা বা ইউনিফায়েড ফাইলসিস্টেম। মাউন্ট করা বিভিন্ন পার্টিশনের ছবিটা, মানে আজকের দ্বিতীয় ছবিটা দেখুন, এর মধ্যে বিভিন্ন আলাদা আলাদা পার্টিশন মাউন্ট করা আছে বিভিন্ন আলাদা আলাদা ডিরেক্টরিতে। যেমন '/dev/hdb5' পার্টিশন আছে '/mnt/arkive' ডিরেক্টরিতে, '/dev/hda6' পার্টিশন আছে '/mnt/slackware' ডিরেক্টরিতে, আর '/dev/hda1' এবং '/dev/hda5' পার্টিশন দুটো মাউন্ট করা আছে '/mnt/windows' ডিরেক্টরির ভিতর '/mnt/windows/c' এবং '/mnt/windows/d' এই দুই সাবডিরেক্টরিতে। এই পার্টিশনগুলো এই এই ডিরেক্টরিতে মাউন্ট করা থাকতে পারে আবার নাও-পারে, এমনকি অন্য কোনো ডিরেক্টরিতেও মাউন্ট করা যেতে পারে, সেটা আমরা একটু বাদেই দেখব। কিন্তু পার্টিশন মাউন্ট করা থাকুক আর নাই-থাকুক, ডিরেক্টরিটা কিন্তু বানানো আছেই। সেটাই আমরা দেখিয়েছি যেন নলপথ আর তা আটকে দেওয়ার স্টপকক দিয়ে। যেন এক একটা পার্টিশনের এক এক টুকরো ডিস্কভূমি একটা আলাদা আলাদা রসদ। আলাদা আলাদা চৌবাচ্চায় আছে। তাদের সঙ্গে মূল চৌবাচ্চার যোগাযোগ ঘটে হায়েরার্কিকাল ডিরেক্টরি পথ বেয়ে, যা খোলা থাকে যখন পার্টিশনটা মাউন্টেড, আর যখন মাউন্টেড নেই তখন বন্ধ। মূল '/dev/hdb3' চৌবাচ্চাটা দেখুন '/' ডিরেক্টরি। তার সঙ্গে '/boot' ডিরেক্টরিতে '/dev/hdb1' পার্টিশন রয়েছে নলপথ দিয়ে সংযুক্ত। কিন্তু সেই নলে কোনো স্টপকক নেই, মানে এই পার্টিশনটা সবসময়েই মাউন্টেড আছে। আর এর ঠিক নিচেই রয়েছে '/dev/hdb2' পার্টিশন। কিন্তু তার কোনো নলপথ মানে ডিরেক্টরিপথ নেই। নলের দাগদুটো তাই, দেখুন, ভাঙা-ভাঙা আঁকা হয়েছে। এবং এর পাশে লেখা 'সোয়াপ'। এর মানে এই সোয়াপ পার্টিশনটাও সবসময়েই সংযুক্ত, কিন্তু সেটা একটা সম্পূর্ণ অন্যরকম ব্যবস্থা, স্বাভাবিক ফাইলব্যবস্থা বা ফাইলসিস্টেম থেকে একদম স্বতন্ত্র। এখানে আমরা ফাইলব্যবস্থা বলতে বোঝাচ্ছি ফাইল লেখা বা রাখা বা পড়ার রকম, যার নানা আলাদা আলাদা ব্যবস্থা হয়, আমরা আগেই বলেছি, ইএক্সটিটু রাইজারএফএস এক্সএফএস ইত্যাদি। কিন্তু সেই সবগুলো ফাইলব্যবস্থা এক দিকে, আর সোয়াপ ব্যবস্থা অন্য দিকে। সোয়াপ একদম আলাদা, তার কারণটা তো আপনারা জানেনই, সোয়াপ ফাইলের জন্মই হয় মৃত্যুর জন্যে, নিজে মরে গিয়ে অন্য ফাইলকে বা ফাইলব্যবস্থাকে বদলে যাবে বলে, অস্থায়ী ভাবে স্মৃতিকে লিখে রাখার একটা জায়গা সোয়াপ-ফাইল, আমরা বারবার বলেছি — সোয়াপ ফাইল হল র‍্যামের সহচর, র‍্যামকে বাড়িয়ে তোলে। তাই এই পার্টিশনের ফাইল লেখা রাখা পড়ার কায়দা অন্যরকম হবেই।

তার মানে, একটা মূল পার্টিশন যা মাউন্ট হবে '/' ডিরেক্টরিতে, আর একটা সোয়াপ পার্টিশন — এই দুই রকমের দুটো পার্টিশনে দুই ধরনের ফাইল-ব্যবস্থা, এটা একটা গ্নু-লিনাক্স সিস্টেমের ন্যূনতম প্রয়োজনীয় শর্ত। এরপর কী কী পার্টিশন থাকবে কী কী ফাইলব্যবস্থায় সেটা আপনার ইচ্ছের এবং প্রয়োজনের উপর নির্ভর করছে। প্রথমটাকে সচরাচর বলা হয় রুট পার্টিশন আর দ্বিতীয়টাকে সোয়াপ পার্টিশন। এমনকি খুব বিদঘুটে বিতিকিচ্ছিরি একটা অবস্থায়, যাকে ইংরিজিতে বলে পরিস্থিতি, এমনকি এই সোয়াপ পার্টিশনও না-রাখা যায়। কিন্তু রুট পার্টিশন না-থাকা মানে সিস্টেমটাই না-থাকা। গ্নু-লিনাক্স একটা সিস্টেমে এই রুট পার্টিশনেই থাকে সিস্টেমের গোটা কাঠামোটা। রুট ডিরেক্টরি এবং তার থেকে বেরোনো সমস্ত সাবডিরেক্টরির কাঠামোটা মনে করুন, ছ নম্বর দিনে এসেছিল, যার কোথায় কী থাকবে সেই ছকটা, যার নাম লিনাক্স ফাইলসিস্টেম হায়েরার্কি, যেটা আপনাদের বারবার বলে আসছি, ওই সামনে, ওই সামনে, কিন্তু কিছুতেই আর এসে পৌঁছচ্ছি না। আসলে লেখার আগে আমি বুঝতেও পারিনি, এক একটা ছোট ছোট জায়গা বলে আসতে এতটা করে লিখতে হবে। যাকগে, এই রুট পার্টিশনের মধ্যে ওই হায়েরার্কিকাল ডিরেক্টরি ব্যবস্থাটা থাকে, থাকে সমস্ত অবশ্য প্রয়োজনীয় ফাইল এবং প্রোগ্রাম।

সোয়াপ পার্টিশন, যা অবশ্যই থাকা উচিত একটা সিস্টেমে, কাজ করার সুবিধের জন্যেই, মূলত কারনেলের কাজের সহায়তা করে। এক আর দুই নম্বর দিনে বিভিন্ন ধরনের মাল্টিপ্লেক্সিং-এর আলোচনার সূত্রে আমরা যে একত্র চলমান অগণ্য প্রসেস বা প্রক্রিয়ার কথা বলেছি, সেই প্রক্রিয়াগুলোর প্রতিটির গতিশীলতাকে, কাজকে, কাজ করতে গিয়ে সিস্টেমের নানা রসদের ব্যবহারকে চূড়ান্ত ভাবে নিয়ন্ত্রণ করে কারনেল। এই নিয়ন্ত্রণ এবং সীমিত রসদকে সুবিধেজনক ভাবে বিভিন্ন প্রক্রিয়ার মধ্যে ভাগবাঁটোয়ারার কাজে কারনেলকে সহায়তা করে সোয়াপ। অনেকগুলো সমান্তরাল

প্রক্রিয়ার ভারে সিস্টেম যখন ভারাক্রান্ত, সিস্টেমের মেমরির পিঠ যখন বেঁকে গেছে বোঝা তুলতে তুলতে তখন তার একটু দম নেওয়ার ব্যবস্থা করার জন্যে কারনেল কিছুটা বোঝা অস্থায়ী ভাবে নামিয়ে রাখে এই সোয়াপে। সোয়াপ শব্দটার আভিধানিক অর্থ একটা কিছুর সঙ্গে অন্য একটা কিছুকে বদলানো। এখানেও ঠিক তাই ঘটে, র‍্যামের একদম জ্যান্ত চলমানতা থেকে সোয়াপের অস্থায়ী আরামে পৌঁছয় একটা প্রক্রিয়া। তারপর যেই ফুরসত আসে, আরামে বদলানো প্রক্রিয়া আবার দৌড় করাতে শুরু করে কারনেল, সোয়াপ থেকে তাকে তুলে র‍্যামে ফেরত এনে। এই সোয়াপ পার্টিশনের এই বিশেষ ধরনের ফাইলব্যবস্থার বৈশিষ্ট্য হল এই যে একে সরাসরি কোনো ব্যবহারকারী পড়তে পারেনা, এমনকি রুট ইউজারও না। সোয়াপকে ব্যবহার করে শুধু কারনেল।

এবার, সিস্টেমে বসে আপনি সোয়াপ সম্পর্কে আর একটু কিছু জানতে চাইলে, তার উপায় কী? মনে আছে? প্রথমে দিন '`man -k swap`'। স্ক্রিনে অনেকটা লেখা ফুটে উঠলে তাকে ফাইল বানিয়ে পড়ার কোনো উপায় আছে? এরকম কিছু শুনেছেন কোথাও? যাকগে, এখানে সে সমস্যা নেই, সোয়াপটা মোটামুটি ভদ্রগোছেরই আছে। মাত্র আট দশটা লাইন। এখানে তার কয়েকটা লাইন তুলি।

```
mkswap(8)  - set up a Linux swap area
swapon(2)  - start/stop swapping to file/device
swapoff(2) - start/stop swapping to file/device
swapon(8)  - enable/disable devices and files for paging and swapping
swapoff(8) - enable/disable devices and files for paging and swapping
```

ওই গোটা দশেক লাইনের থেকেও কয়েকটা ছেঁটে দিলাম, এত বেশি টেকনিকাল সেগুলো। এবার দেখুন তো, নিজে পড়ে, কী আন্দাজ পাচ্ছেন কমান্ডগুলো সম্পর্কে? 'mkswap' স্বাভাবিক ভাবেই, সোয়াপফাইল বানায়, মানে যে পার্টিশনটাকে আপনি সোয়াপ পার্টিশন বানাতে চান, সেই পার্টিশনে সোয়াপ ফাইল লেখার বানানোর ব্যবহার করার মত করে নিজস্ব স্বতন্ত্র ফাইলব্যবস্থা বানিয়ে তোলে 'mkswap'। সচরাচর এটা চালু প্রথা যে নতুন ব্যবহারকারীর গ্নু-লিনাক্স ইনস্টলেশন করে দেয় তার চেনাজানা কোনো লিনাক্সী। এমনকি আপনি নিজেও যদি ইনস্টল করে থাকেন, তাহলেও, ইনস্টল করার সময়, আপনি আসলে এই কমান্ডটাই ব্যবহার করেছেন, না-জেনে। ইনস্টলেশনের সময় যখন দেখিয়েছে, কোথায় সোয়াপ পার্টিশন রাখব, কতটা তার সাইজ রাখব, তখন আপনি চালু প্রথা মেনে, মোটামুটি র‍্যামের ডাবল, সাইজ দিয়েছেন, কোথায় বানাবে বলে দিয়েছেন, বা এমনকি আপনার হয়ে সিস্টেমই সেটা করেছে, নিজেই মোটামুটি একটা গড় ব্যবস্থা ছকে নিয়েছে, তারপর সেই অনুযায়ী ইনস্টল করেছে। সিস্টেম নিজেই করুক, অন্য কেউ করে দিক, আপনি করে থাকুন, শেষ অব্দি ব্যবহার হয়েছে এই 'mkswap'। সামনে আপনি কোনো ছবিওলা ঝিনচাক দেখানোর দাঁত দেখেছেন — হলুদ লাল নীল ধূসর, কোনোটা ইএক্সটিটু, কোনোটা রাইজার, কোনোটা ফ্যাটথার্টিটু, কোনোটা পদিপিসির বর্মিবাক্স, এই রোগা হচ্ছে, এই মোটা হচ্ছে, এই রং পালটাচ্ছে। সেই ফ্রন্ট-এন্ডের পিছনে খাওয়ার দাঁত মানে 'mkswap' দিয়েই হার্ডডিস্ক ভূমিটাকে চিবিয়ে চিবিয়ে সোয়াপ ফাইল লেখার মত করে সাইজ করেছে সিস্টেম। আর কোন পার্টিশনে আপনি সোয়াপ বানাবেন সেটা সিস্টেমকে দিয়েছেন 'swapon' ব্যবহার করে, আপনি হয়ত ইঁদুরের পেট টিপেছেন, কিন্তু ব্যাক-এন্ডে ছিল 'swapon'। তেমনি আপনি যদি কোনো পার্টিশনে সোয়াপ ফাইল আর না-রাখতে চান, সেটাকে অন্যকাজে ব্যবহার করতে চান, তখন, আপনার ইঁদুরের নাড়িভুড়ি বেয়ে দৌড় করেছে 'swapoff'।

এবার আপনার এই 'হয় হয় কিন্তু Zানতে না-পারা' ক্যালিটা সরাসরি কমান্ড প্রম্পটে শো করে দিয়ে নিজের ইগোকে একটু পিঠ-চুলকে দিতে পারেন, কিন্তু, দেখবেন, ভালো করে পার্টিশন ছক না-বুঝে এইসব করতে গিয়ে সিস্টেমকে কাতুকুতু দিয়ে বসবেন না, গোটাটা নেড়ে ঘেঁটে গিয়ে একটা সমূহ কেলেংকারি। আর কমান্ডগুলো সম্পর্কে ভালো করে জানবেন কী করে? মরুঅঞ্চলের উদ্ভিদের বাংলার পলিমৃত্তিকায় কৃষির সাফল্য বিষয়ে গবেষণা নিয়ে কী একটা গান আমাদের কানে এসেছিল না?

ও, একটা কথার উত্তর দিন তো, এই তালিকাটায় 'swapon' আর 'swapoff' কমান্ডদুটোর দুবার করে দুটো আলাদা নম্বরের ম্যানপেজের কথা বলার মানে কী? দেখুন, দুটো কমান্ডই দুবার করে আছে একবার '2' আর একবার '8'। তার মানে একটা দেখতে হবে '`man 2 swapon`' বা '`man 2 swapoff`' কমান্ড দিয়ে, আর একবার দেখতে হবে '`man 8 swapon`' বা '`man 8 swapoff`' কমান্ড দিয়ে। কিন্তু কেন? যদি না বুঝতে পারেন ছয় নম্বর দিনের

গোড়ার দিকটা একবার উল্টে আসুন। গোটা পুরোনো আলোচনাটা আপনার মাথায় একদম হাতেগরম থাকা চাই। সচরাচর টেক্সটগুলোয় যা হয়, ঠিক গঠনের নিয়ম মেনে প্রসঙ্গগুলো পরপর আসে, তাতে একটু বোরিং হয় ঠিকই, কিন্তু একটা সুবিধাও থাকে, এগোনোর পিছনোর, আটকে গেলে পিছিয়ে গিয়ে বুঝে-আসার। একদম বিষয়-নির্ভর বা অবজেকটিভ পাঠ। আর এই লেখাটা হচ্ছে তার বিপরীত ধরনে, বিষয়ী-নির্ভর বা সাবজেকটিভ পাঠ বলা যায়। যেন একজন জ্যান্ত কেউ সিস্টেমটা বুঝতে বুঝতে চলেছে, যেখানে যেখানে তার যেটুকু আটকে যাচ্ছে সেটুকু সে আলোচনা করে নিচ্ছে, ওই জ্যান্ত ব্যবহারকারীটা বেশি উপস্থিত ব্যবহারের বিদ্যাটার চেয়েও। তাই এখানে ব্যবহারকারীর দায়িত্ব একটু বেশি এসে পড়েই, গোটাটা নিজের মাথায় তুলতে তুলতে যাওয়ার। এখানে আটকে গেলে আপনি তো খুঁজেই পাবেন না প্রসঙ্গটা কোথায় এসেছে। ঠিক আমাকে যতটা গোটা লেখাটা মাথায় রাখতে হচ্ছে আপনাকেও তাই করতে হবে, আমি খেয়ে-না-খেয়ে লিখে চলেছি, আর আপনি চুয়িংগাম চিবোতে চিবোতে পড়ে ফেলবেন, সেটা চলবে না।

৩।। নানা ধরনের ফাইলব্যবস্থা

আগের সেকশনে আমরা বললাম রুট আর সোয়াপ — ফাইলব্যবস্থার এই দুটো বড় বিভাগের কথা। এবার এই রুট পার্টিশনের ফাইলব্যবস্থাটা ঠিক কী রকম হবে? তার নানা প্রকারভেদ আছে। সোয়াপ ফাইলব্যবস্থায় যা নেই। আমার মেশিনের একই সোয়াপ পার্টিশন সুজে আর স্ল্যাকওয়ার দুজনেই সমানভাবে ব্যবহার করে, যখন যেটা দিয়ে বুট করি। আগে যখন সুজে স্ল্যাকওয়ার আর রেডহ্যাট তিনটেই ছিল, তিনটেরই তাই। সোয়াপ ফাইলব্যবস্থার প্রকারভেদ হয়না। কিন্তু রুট পার্টিশনের হয়। আর এছাড়াও নানা পার্টিশন থাকে, নানা আলাদা আলাদা ফাইলব্যবস্থায়, আমরা আগেই দেখেছি। সিস্টেমের নিরাপত্তা, ব্যবহারের সুবিধার জন্যেও নানা পার্টিশন রাখতে হয়, আমরা বলেছি। আপনার গ্নু-লিনাক্স সিস্টেমে একটা বিশেষ মুহূর্তে মোট কত রকমের ফাইলব্যবস্থা ব্যবহৃত হচ্ছে, সেটা জানার একটা উপায় হল, ‘`cat /proc/filesystems`’।

এই ‘/proc’ ডিরেক্টরিটা আমরা ঐক্যবদ্ধ ফাইলসিস্টেমের ডিরেক্টরি কাঠামোর ছবিতে দেখেছি, এখানে কী থাকে সেটায় আমরা একটু বাদে আসব গ্নু-লিনাক্স ফাইলসিস্টেম হায়েরার্কির আলোচনায়। এই যে একটা বিশেষ মুহূর্তের কথা বললাম, সেটা খুব গুরুত্বপূর্ণ। ধরুন, আপনার সিস্টেমে একটা পার্টিশন আছে যেখানে ফাইল লেখার রাখার পড়ার ব্যবস্থাটা হল রাইজারএফএস, এবার সেই পার্টিশনটা আপনি সবসময় মাউন্ট করেন না, যখন দরকার পড়ে মাউন্ট করে নেন। আমার সিস্টেমে ‘/dev/hdb5’ আর ‘/dev/hda6’ পার্টিশনদুটো যেমন, আমার যখন দরকার পড়ে আমি মাউন্ট করে নিই ‘/mnt/arkive’ আর ‘/mnt/slackware’ ডিরেক্টরিতে। মাউন্ট কখন করবে, আপনাআপনি সিস্টেম একটা পার্টিশনে মাউন্ট করে নেবে কিনা — এসব নিজের মর্জিমত ঠিক করে নেওয়া যায়, ‘/etc/fstab’ বলে একটা ফাইল মারফত, মাউন্টের আলোচনায় আমরা সেকথায় আসব। এছাড়া আর কোনো পার্টিশনে রাইজারএফএস নেই। এবার, যখন আমার সিস্টেমে এই পার্টিশনদুটো মাউন্ট করা নেই, তখন আমি ‘`cat /proc/filesystems`’ কমান্ড দিলে ও কোনো রাইজারএফএস ওর তালিকায় দেখায় না, কিন্তু মাউন্ট করা থাকলে দেখায়।

এবার আমার সিস্টেমে সবগুলো পার্টিশন মাউন্ট করা অবস্থায় আমি ‘`cat /proc/filesystems`’ দিয়ে যে তালিকাটা পেলাম সেটা এখানে তুলে দিই। অবশ্যই এই রকম চার স্তম্ভে আমি পাইনি, সেটাকে করে নিতে হয়েছে ‘’ বলে একটা কমান্ড দিয়ে। ‘`cat /proc/filesystems|pr -4>filesystems`’ কমান্ড দিয়ে প্রথমে তালিকাটাকে পাইপ করা হয়েছে ‘pr’-এর কাছে, ‘pr’ তালিকাটাকে চার স্তম্ভে স্তম্ভিত অবস্থায় রিডাইরেক্ট করেছে ‘filesystems’ নামের ফাইলে। ‘pr’ আসলে প্রিন্ট সংক্রান্ত কমান্ড, আগে ‘lpr’ বলে একটা কমান্ড হত, লাইন প্রিন্ট করার কাজে ব্যবহারের, কিন্তু এই সব কাজে বেড়ে ব্যবহার করা যায়, পাতার নম্বর, কলাম, রো, হেডার, ফুটার, সব ঠিক করে দেয় একটা টেক্সট ফাইলে। নিজে ব্যবহার করে দেখুন। সেই ফাইল থেকে তুলে দিচ্ছি, একটু ছাঁটকাট করে। আপনারা যখন ম্যানপেজ পড়ছেন, তখন তাকেও রিডাইরেক্ট করে একটা ফাইলে এনে সেটাকে এইভাবে ‘pr’ দিয়ে ফরমাট করে নিতে পারেন, পড়ার সুবিধে হয়। এছাড়া ম্যানপেজ থেকে ওয়েবপেজ বা ‘*.html’ ফরম্যাট বা অন্য কোনো ফরম্যাটে বদলে নেওয়ার অনেক উপায় আছে। আমি যেটা ব্যবহার করি সেটা ‘rman’। ধরুন আপনি ‘ls’-এর ম্যানপেজ পড়তে চাইছেন। তাকে রিডাইরেক্ট করে ফাইল বানিয়ে নিতে পারেন ‘`man ls>lsman`’। একে ‘pr’

দিয়েও ফরমাট করে নিতে পারেন। বা ‘rman’ দিয়ে করলে, ‘`man ls|rman -f html>lsman.html`’। এতে ‘ls’ কমান্ডের ম্যানপেজটাকে ‘rman’ একটা ওয়েবপেজ করে দেবে যার নাম ‘lsman.html’। একে আপনি আপনার ইন্টারনেট ব্রাউজার দিয়ে পড়তে পারবেন। কমান্ড প্রম্পটে ব্যবহার করার জবর দুটো টেক্সট মোড ব্রাউজার হল ‘links’ আর ‘lynx’। আরো কী কী আছে, মনে করতে পারছি না। এই ব্রাউজারদুটো দিয়ে চাইলে ছবিসহ একদম স্বাভাবিক চেহারাতেও ব্রাউজ করা যায়, তখন আলাদা করে ‘`-graphics`’ অপশান দিতে হয়।

আমি তো অনেকসময়ই নেটে ব্রাউজ করি টেক্সট মোডে। এতে সুবিধে হল, প্রচুর গাম্বাট লোক তাদের ওয়েবপেজে গাড়লের মত দুম্বো দুম্বো সব গ্রাফিক্স আর জাভাস্ক্রিপ্ট দিয়ে রাখে, তার আর কী, খরচ তো আপনার, লোড হতে প্রচুর সময় লাগে। ‘links’ বা ‘lynx’ টেক্সট মোডে নিট দরকারি জিনিষটুকু এনে দেয়, দরকার পড়লে জাস্ট একটা ‘d’ মেরে ডাউনলোড করে নাও। আর আপনার ফরম্যাট করা ম্যানপেজগুলো বা যে ম্যানপেজের সংগ্রহ এমনিতেই দেওয়া আছে আপনার সিস্টেমে তাদেরও পড়তে পারেন দিয়ে। আপনি, কমান্ড প্রম্পটে থাকলে তো ভালোই, যদি এই মুহূর্তে এক্স-উইনডোজেও থাকেন, অসুবিধে নেই, একটা টার্মিনাল বা একটা কনসোল খুলুন, নেটে কানেক্ট করে কমান্ড দিন ‘`links www.geocities.com/ddipankardas/`’। আর যদি ব্যাকগ্রাউন্ডে জিএলটির লোগোর অনবদ্য ছবিটা সহ — আমি করেছি না? — পাতাটা দেখতে চান, তখন জাস্ট একটা ‘`-g`’ যোগ করে দিন কমান্ডটার সঙ্গে। ছবিদুটো কুচো সাইজের বললেও কম বলা হয়, ‘glt-logo.png’ আর ‘glt-bkg.png’ দুটোই দুই কেবির কম। এটা ইয়াহু জিওসিটিতে মধ্যমগ্রাম জিএলটির সাইট, বানানো এখনো শেষ হয়নি, উপরের কয়েকটা মাত্র লিংক কাজ করছে। এই লেখাটা শেষ হলে, তথাগত অশেষ সঙ্কর্ষণ ওদের সব পোকাবাছা হয়ে গেলে সবগুলো চ্যাপ্টারই থাকবে এখানে, পিডিএফে। ‘links’ কমান্ড দেওয়ার পরে যদি দেখায় ‘`command not found`’, তার মানে, আপনার সিস্টেমে ‘links’ ইনস্টল করা নেই, জাস্ট ইনস্টলেশন সিডিটা দিয়ে ইনস্টল করে নিন। ‘rman’-ও তাই। যাকগে, এবার আসা যাক ফাইলসিস্টেমের তালিকায়। ‘pr’ কি সুন্দর নিজে নিজেই আজকের তারিখ সময় আর পাতার নম্বর করে দিয়েছে দেখুন।

```
2004-01-07 11:07                                      Page 1

rootfs         tmpfs          minix          usbdevfs
bdev           shm            iso9660        usbfs
proc           pipefs         nfs            reiserfs
sockfs         ext2           devpts         vfat
futexfs        ramfs          xfs
```

এই ফাইলসিস্টেমের নামের তালিকা থেকে দেখুন, কয়েকটা নাম আপনার চেনা, অন্য কয়েকটা নয়। দু-চারটে কথা বলে নেওয়া যাক এদের নিয়ে। শুরু করা যাক গ্নু-লিনাক্সে সবচেয়ে বেশি ব্যবহৃত ডিফল্ট ফাইলসিস্টেমগুলো থেকে। কয়েক বছর আগেও এই বিষয়টা ছিল খুব সরল, ব্যবহারযোগ্য ফাইলব্যবস্থা গ্নু-লিনাক্সে দুই কি ম্যাক্সিমাম তিনের বেশি ছিলনা, মূলত ব্যবহার হত ইএক্সটিটু নিদেন রাইজারএফএস। এখন তা নয়, অনেকগুলো ফাইলসিস্টেম রেগুলার ব্যবহার হয়ে চলেছে অজস্র গ্নু-লিনাক্স সিস্টেমে। ইএক্সটিটু ক্রমে দিগন্তের ওপারে চলে যেতে শুরু করেছে। এমনকী পরিবর্তনের গড়িমসির জন্যে পরিচিত রেডহ্যাট লিনাক্সও তার ভার্শন নাইন থেকে ডিফল্ট করে দিয়েছে ইএক্সটিথ্রিকে। ম্যানড্রেক সুজে এগুলোকে তো ছেড়েই দিন। এমনকী বিশুদ্ধতাবাদী স্ল্যাকওয়ারও তার ইনস্টল সিডির কারনেলগুলোর ভিতর একটা এক্সএফএস কারনেলও দিয়ে দিচ্ছে। কারনেল ভারশন ২.৪ থেকে গ্নু-লিনাক্স অনেকগুলো ফাইলসিস্টেমকে ব্যবহার করার সুযোগ দেয়। তার অনেকগুলো ফাইলসিস্টেমের খুঁটিনাটিই দেখবেন দেওয়া আছে আপনার সিস্টেমে, ‘/usr/src/linux/Documentation’ ডিরেক্টরির মধ্যে। টেক্সট ফাইলে।

এখানে আমরা একটা ফাইলসিস্টেমেরও গঠনের খুঁটিনাটি জটিলতাতেও যাবনা, সে এক বীভৎস বিরাট ব্যাপার, এখানে কিছুটা দেওয়ার প্ল্যানে আমি অনেকটা পরিশ্রম করলাম কয়েকদিন ধরে। সঙ্কর্ষণও একটা ভালো লিংক দিয়েছে, আইবিএম সাইটে, লিনাক্স ফাইলসিস্টেম সংক্রান্ত। ড্যানিয়েল রবিনস-এর দশটা নিবন্ধের একটা সিরিজ, http://www-106.ibm.com/developerworks/library/l-fs.html, লিনাক্স ফাইলসিস্টেম নিয়ে। আর একটা খুব ভালো লিংক জুয়ান স্যান্টোস ফ্লোরিডোর লেখা, জারনাল ফাইল সিস্টেম নিয়ে, লিনাক্স গেজেটে। তার লিংকটা হল, http://www.linuxgazette.com/issue55/index.html। নেমসিস-এর (Namesys) নিজের সাইটে, মানে যাদের

রাইজারএফএস, রাইজারের উপর খুব ভালো একটা ডকুমেন্ট আছে। ঠিক এসজিআই-এ (SGI) যেমন আছে তাদের নিজেদের তৈরি এক্সএফএস নিয়ে। আমার ইচ্ছে ছিল বেশ কিছুটা খুঁটিনাটি নিয়ে এগুলোকে ধরা, শেষে নিজেরই কাপড়চোপড় খুলে যাওয়ার যোগাড়, একটাকে বুঝতে আর একটা, তাকে বুঝতে আর একটা, তার উপর খুব বেশি সময় নেটে থাকতে গেলে নিজেরই অপরাধবোধ হয় একটা, সংসার তো চালাতে হবে। আর এই লেখাটাও প্রায় শেষের দিকে চলে আসছে, খুব বেশি লম্বা করার ইচ্ছেও নেই। এখানে আমরা তাই খুব শর্টে সারব, জাস্ট এক একটা ফাইলসিস্টেমের সুবিধে অসুবিধে নিয়ে, কী কাজের উপযোগী, ইত্যাদি। একটা কথা মাথায় রাখবেন, নানা ধরনের ফাইলসিস্টেম লাগে নানা ধরনের কাজের জন্যে। এমন একটা ফাইলসিস্টেমও নেই যা সব কাজ সবচেয়ে ভালো ভাবে করতে পারে। সবচেয়ে নিরাপদ এবং সুস্থির ফাইলসিস্টেমেও প্রচুর গন্ডগোল তৈরি হয়, তাদের সামলাতে হয়, তাই সবসময়ই ব্যাকআপ করা উচিত নিজের তথ্যের, নিয়মিত ভাবে।

৩.১।। ইএক্সটিটু

গ্নু-লিনাক্স জন্মানোর একদম প্রথমতম ইতিহাস থেকেই প্রায় চালু ইএক্সটিটু (Ext2)। ইএক্সটিটু এসেছে তার আগেকার এক্সটেন্ডেড ফাইলসিস্টেম ব্যবস্থা থেকে, নামের আত্মীয়তাটা খেয়াল করুন। এক্সটেন্ডেড ফাইল সিস্টেম (Extended-File-System) সিনে এসেছিল এপ্রিল ১৯৯২-এ। কারনেল ভার্শন ০.৯৬সি-র সময়। এরপর বারবার বদলাতে থেকেছে। এই বদলের সূত্র ধরেই এসেছিল ইএক্সটিটু এবং গ্নু-লিনাক্সের সবচেয়ে জনপ্রিয় ফাইলসিস্টেম হয়ে উঠেছিল একটানা বেশ কয়েকবছর ধরে। জার্নালিং ফাইলসিস্টেম মানে জার্নাল ব্যবহার করে এমন ফাইলসিস্টেম আসার পরই ধীরে ধীরে সিন থেকে সরে যেতে শুরু করল ইএক্সটিটু। জার্নাল ব্যাপারটা নিয়ে বিশদ আলোচনার লিংক তো আগেই দিয়েছি, এই লেখাটা তার জন্যে খুব একটা উপযুক্ত জায়গাও নয়, এবং আমিও আলোচক হিশেবে আদৌ খুব উপযুক্ত নই, এত বেশি টেকনিকাল সেটা। তবু ব্যাপারটা কী সেটা একটু বলে নিই।

একটা ফাইলসিস্টেমের জার্নাল বলতে বোঝায় একটা ছক, একটা কাঠামো। এই ছকটাও রাখা থাকে ডিস্কের ওই পার্টিশনেই, যে পার্টিশনের ফাইলসিস্টেমকে জার্নাল করা হচ্ছে। ছক বা কাঠামোটা হল একধরনের একটা নথি বা লগ, যেখানে সমস্ত বদলগুলো লিপিবদ্ধ হয়। ইনফ্যাক্ট বলা ভাল, বদলগুলো হওয়ার আগেই লিপিবদ্ধ হয়। ফাইলসিস্টেম তার নিজের মেটাডেটায় কোনো বদল ঘটানোর আগে সেই বদলের হিশেবনিকেশ এই জার্নালে লিখে রাখে। মেটাডেটা বলতে বোঝায় একটা ফাইলসিস্টেমের আভ্যন্তরীন তথ্যের কাঠামো। যে কাঠামোটার কাজ এটা নিশ্চিত করা যে তথ্যের এবং তথ্যের সজ্জাটার থোবড়া বিলা হয়ে যায়নি, হুলিয়া বদলে যায়নি, ঘেঁটে যায়নি সেগুলো। এই 'মেটা' কথাটা উত্তরআধুনিক সংস্কৃতিচর্চারও একটা খুব প্রিয় টার্ম। মেটান্যারেটিভ, মেটাটেক্সট — ন্যারেটিভ বিষয়ে ন্যারেটিভ, টেক্সট বিষয়ে টেক্সট। আমি অনেকদিন আগে 'নিরন্তর প্রব্রজ্যায়: দ্বিতীয় খসড়া' নামে একটা উপন্যাস লিখেছিলাম, সেটা এই অর্থে মেটানভেল। একটা উপন্যাস কী করে লিখব তাই নিয়ে একটা উপন্যাস। যে কোনো কিছুতেই এটা ঘটতে পারে। কবিতা নাটক ফিল্ম। এখানে মেটাডেটা সেই রকম ডেটা বিষয়ে ডেটা, তথ্য বিষয়ে তথ্য। প্রত্যেক ফাইলসিস্টেমেরই নিজের একটা মেটাডেটা থাকে, এবং মেটাডেটা লেখার একটা রকম থাকে। মেটাডেটার কাঠামোর তফাত একটা ফাইলসিস্টেম থেকে আর একটা ফাইলসিস্টেমের তফাতের একটা বড় জায়গা। মেটাডেটাকে বদলে যেতে না-দেওয়া — একটা ফাইলসিস্টেমের জরুরি কাজগুলোর মধ্যেও সবচেয়ে জরুরি। কারণ, ভেবে দেখুন, মেটাডেটা বদলানো মানেই আপনার সিস্টেম আর ওই পার্টিশনের ওই ফাইলসিস্টেমে রাখা তথ্যের সজ্জাকে কাঠামোকে তথ্যকে চিনতে পারবেনা। তথ্যগুলো তখন আছে এবং নেই। মানে নেই। ওরকম সন্ধ্যাভাষার অস্পষ্টতা নিয়ে সাইকোলজি চলে, কাব্য চলে, হার্ডডিস্কের প্ল্যাটার আর হেড চলেনা।

জার্নালাইজড ফাইলসিস্টেমের জার্নালের কাজ এই মেটাডেটায় কোনো বদল ঘটানোর আগে লিখে রাখা। সিস্টেম বুট করার সময় ফাইলসিস্টেম যে চেক হয়, বা কোনো গন্ডগোল ঘটলে, বিদ্যুৎবিভ্রাট ইত্যাদি, যখন চেক করা হয়, আগেই বলেছি আমরা, সেই চেকের বা নিরীক্ষার কাজে সময়ের প্রয়োজনটা ভয়ানকভাবে কমিয়ে দেয় জার্নাল। এমনিতে ভাবুন, প্রতিটি সিলিন্ডারের ট্র্যাকের কাঠামো ধরে ধরে তথ্য এবং তথ্যের সজ্জাকে মেলানোর কাজটা কমে গিয়ে দাঁড়াচ্ছে জাস্ট জার্নাল মেলানোর কাজে। গোটা ডিরেক্টরি সিস্টেম ফাইল সিস্টেমটা আর মেলাতে হচ্ছেনা। শুধু জার্নালটা আর একবার চালিয়ে দেওয়া হচ্ছে। এই জার্নাল ব্যবহার করা ফাইলসিস্টেম আসার পরেই ইএক্সটিটু ফাইলসিস্টেমের জনপ্রিয়তা কমে যেতে শুরু করল। আসলে জার্নাল ব্যবহার করাটা ফাইলসিস্টেম চেকের সময়টাকে

এমন অবিশ্বাস্য ভাবে কমিয়ে দেয়। কিন্তু এখনো একটা বিরাট সংখ্যক লিনাক্সীর কাছেই ইএক্সটিটু খুব প্রিয় একটা ফাইলসিস্টেম। তার একটা বড় কারণ এই যে, এটার সঙ্গে দীর্ঘদিনের অভ্যস্ততা ছাড়াও, এত বেশি মেশিনের এত বেশি সিস্টেমে এত বেশি ব্যবহারকারী ইএক্সটিটুকে ব্যবহার করেছেন যে এর মত বারবার পরীক্ষিত ফাইলসিস্টেম আর হওয়া সম্ভব নয়। স্বাভাবিক অবস্থায়, সিস্টেম অফ হওয়ার সময়, কারনেল সমস্ত বাফারকে তার নিজের নিজের জায়গায় সঠিকভাবে লিখে দিয়ে পার্টিশন থেকে বেরিয়ে আসে, মানে পার্টিশনটাকে নিজের সিস্টেমের বাইরে বার করে দেয়। একে বলে আনমাউন্ট করা। এক এক করে কারনেল প্রতিটি পার্টিশন থেকে আনমাউন্ট করে, এবং এই পক্রিয়ার শেষে সে নিজেই নিজের কাছ থেকে ছুটি নেয়, আমরা স্ক্রিনে ফুটে উঠতে দেখি

`Sending all processes the TERM signal...`

`Sending all processes the KILL signal...`, মেশিন অফ হয়।

বিদ্যুৎবিভ্রাট বা অন্য কোনো কারণে, ভুল ভাবে মেশিন বন্ধ করার কারণে, বা অন্য কোনো গোলযোগে, যখন স্বাভাবিক আনমাউন্ট ঘটেনা, মানে, সঠিকভাবে বাফারগুলোকে লিখে সমস্ত জরুরি তথ্যকে তার নিজের জায়গায় রেখে সঙ্গতভাবে বেরিয়ে আসতে পারেনা কারনেল, তখন পরের বার বুট হওয়ার সময় ফাইলসিস্টেম চেকিং শুরু হয়। আমরা 'fsck' কমান্ডটার কথা বলেছিলাম, এর ইএক্সটিটু ভার্শনটার নাম 'fsck.ext2'। সেই 'fsck.ext2' তখন চালু হয়। তার কাজ গোটা মেটাডেটাকে পুনঃসংস্থাপিত করা তার সঠিকতায়। এইটা করতে গিয়ে তাকে গোটা ফাইলসিস্টেমটাই তন্নতন্ন করে খুঁটিয়ে খুঁটিয়ে পরখ করতে হয়, শুধু মেটাডেটার বিটগুলোর সাম্প্রতিকতম বদলটাকে বিশ্লেষণ করলেই কাজ হয়না। 'fsck.ext2' বকেয়া থেকে যাওয়া তথ্য বা ডেটা-ব্লকগুলোকে একটা 'lost+found' নামের ডিরেক্টরিতে তুলে রাখে। তার মানে, প্রচুর প্রচুর সময়, জার্নালের লগটা চালিয়ে দেখতে যা সময় লাগে তার চেয়ে অনেক বেশি। বড় হার্ডডিস্কে এই কাজে দুতিন ঘন্টাও লেগে যেতে পারে। কিন্তু আবার জার্নাল রাখেনা বলেই মেটাডেটার পরিমাণটা অনেক ছোট হয়, মেমরি অনেক কম লাগে এই ফাইলসিস্টেমের ব্যবহারে। তাই ব্যবহারের সময়, যদি সিস্টেমের র‍্যাম খুব কম হয়, ইএক্সটিটু অনেক তাড়াতাড়ি কাজ করতে পারে অন্য অনেক ফাইলসিস্টেমের চেয়ে।

৩.২।। ইএক্সটিথ্রি

আপগ্রেড করা বা উন্নততর করে তোলাটা কম্পিউটার জগতে, হার্ডওয়ার আর সফটওয়ার দুটোতেই, খুব বড় একটা ইশু। ইএক্সটিটুকে ইএক্সটিথ্রিতে আপগ্রেড করে তোলার কাজটা খুবই সহজ। ইএক্সটিথ্রি (Ext3) আবার জার্নালিত ফাইলসিস্টেম। ইএক্সটিটুর সহজতা এবং জার্নালের দ্রুততা একসাথে পাওয়া যায়। ইএক্সটিথ্রির সঙ্গে অন্যান্য পরবর্তী যুগের জার্নালিত ফাইলসিস্টেমের তফাত এটাই যে ইএক্সটিথ্রি কিন্তু এক্সটেন্ডেড ফাইল সিস্টেমের কাঠামো থেকে সম্পূর্ণ বেরিয়ে যায়নি। নামের ধারাবাহিকতাটা দেখুন। ইএক্সটিথ্রি প্রায় সরাসরি ইএক্সটিটু-র উত্তরাধিকার বহন করে। দুটোর আত্মীয়তার জায়গাটা খুবই পোক্ত। আর এই কারণেই একটা ইএক্সটিটু ফাইলসিস্টেমের উপর খুবই সহজে একটা ইএক্সটিথ্রি ফাইলসিস্টেম বানিয়ে তোলা যায়। এদের একমাত্র গুরুত্বপূর্ণ তফাত ওই জার্নাল।

ইএক্সটিথ্রি-র সুবিধেগুলোর একটা, তাই, খুব সহজে ইএক্সটিটু থেকে বদলে নিতে পারা। ফাইলসিস্টেমের আভ্যন্তরীণ তথ্য এবং তথ্যসজ্জার কোনো গলতি হওয়ার সম্ভাবনা থাকেনা। ইএক্সটিটু-র কোডের উপরেই তৈরি ইএক্সটিথ্রির কোড, আর ডিস্কের ভিতর তথ্য রাখার কৌশল বা মেটাডেটার কাঠামো দুটো ফাইলসিস্টেমেই প্রায় হুবহু এক, তাই ইএক্সটিটুতে কোনো পার্টিশন ফরম্যাট করা থাকলে, এবং পরে সেটা বদলে কোনো জার্নালিত ফাইলব্যবস্থায় যেতে চাইলে, রাইজারএফএস (ReiserFS) এক্সএফএস (XFS) বা জেএফএস (JFS) জাতীয় অন্য কোনো সিস্টেমে যাওয়ার যে ঝামেলা সেটা ইএক্সটিথ্রির বেলায় হয়না। ওগুলোয় যাওয়ার আগে পার্টিশনের গোটা তথ্যের ব্যাকআপ করে নিতে হয়, এবং সেই ব্যাকআপ থেকে ফের শুরু করে গোটাটা বানিয়ে নিতে হয়। সেটা ইএক্সটিথ্রি-তে হয়না। আর ফের গোটাটা বানিয়ে নিতে গিয়ে কিছু ঘাপলা না-ঘটাটাই আশ্চর্য। কোনো একটা লিংক কাজ করছে না, কোনো একটা লাইব্রেরি পাচ্ছেনা, ইত্যাদি। এটা ইএক্সটিথ্রি-র বেলায় হয়না, কারণ আসলে ওই কাঠামোটা তো বদলাচ্ছেই না। আবার ইএক্সটিথ্রি থেকে কখনো ইএক্সটিটু-তে ফেরত যেতে চাইলে সেটাও দিব্য সহজ। একটা পরিচ্ছন্ন রক্তপাতহীন আনমাউন্ট করো ইএক্সটিথ্রি পার্টিশন থেকে, আবার তাকে রিমাউন্ট করো ইএক্সটিটু হিশেবে। আমরা দেখব একটু বাদেই, কী করে এই অপশানগুলো দিতে হয়। অন্য জার্নালিত ফাইলব্যবস্থাগুলোয় তথ্যসততা কায়েম রাখার কায়দাটা

হল শুধু মেটাডেটাকে নিয়ন্ত্রণ করে। মেটাডেটা সবসময়েই অনাঘ্রাত অনাহত থাকবে। কিন্তু মূল ফাইলব্যবস্থার অপরিবর্তনীয়তাটা এতটা বর্মচর্মাবৃত নয়। ইএক্সটিথ্রি-র বেলায় তা নয়, সে তথ্য এবং তথ্য-বিষয়ক-তথ্য, ডেটা এবং মেটাডেটা, দুয়েরই নিরাপত্তার দেখভাল করে। কতটা যত্ন নিয়ে দেখভাল করবে সেটাও বলে দেওয়া যায়, মানে নিজের মত করে বদলে কাস্টমাইজ করে নেওয়া যায়। ডেটা এবং জার্নালকে তুল্যমূল্য করে দিলে, 'data=journal' মোডে সবচেয়ে বেশি নিরাপত্তা, কিন্তু কাজ করে খুব আস্তে। এতে তথ্যের গায়ে কোনো আঁচড় পড়েনা, কারণ তথ্য এবং তথ্যের-তথ্য, ডেটা এবং মেটাডেটা, দুয়েরই জার্নাল রাখা হচ্ছে। বরং 'data=ordered' মোডে ডেটা এবং মেটাডেটা দুয়ের অটুটতাতেও নজর রাখা হয়, কিন্তু জার্নাল রাখা হয় শুধু মেটাডেটার। মেটাডেটার এক একবার বদলের সঙ্গে সম্পৃক্ত যাবতীয় ব্লকগুলোকে সংগ্রহ করে রাখে ফাইলসিস্টেম ড্রাইভার, এক একটা দলে, এক একটা 'ট্রানজাকশন'-এর ('transaction') আকারে। এবং মেটাডেটায় বদল ঘটানোর আগে এক একটা গোটা ট্রানজাকশনকে লিখে ফেলা হয় হার্ডডিস্কের শরীরে। এতে কাজের গতিও ঠিক থাকে, আবার তথ্যের নিরাপত্তাও বজায় থাকে। আর একটা মোড হল 'data=writeback'। এই মোডে মেটাডেটার খাতায় বদলের নথী লিখে ফেলার পর হার্ডডিস্কের মূল ফাইলসিস্টেমে তথ্য লেখা হয়। মোডের নামটা দেখুন, আগে-পরে-টা বুঝতে পারবেন। এই তিন-নম্বর মোডে কাজ সবচেয়ে তাড়াতাড়ি হয়। কিন্তু একটা কোনো বিদ্যুৎবিভ্রাট বা ওই জাতীয় কোনো কেলোর পরে একটা বিদঘুটে ব্যাপার ঘটার সম্ভাবনা থাকে। আপনার কবিতার ফাইলে যেসব লাইন আপনি ইতিমধ্যেই উড়িয়ে দিয়েছেন, সেগুলো ফের পেয়ে যেতে পারেন ফাইলে। কেন ভাবুন তো? আভ্যন্তরীন ফাইলসিস্টেমের তো কোনো হানি হয়নি, সেটা অটুট আছে। এবার, আপনি যে বদল ঘটিয়েছিলেন, সেটা মেটাডেটায় লিখে ফেলার, রাইটব্যাক করার আগেই কেলোটা ঘটেছিল। তাই আপনার বর্জিত লাইনেরা আবার কোরাসে গাইতে শুরু করেছে, লাইনেই ছিলাম বাবা। যদি নিজের থেকে কোনো পছন্দ না দিয়ে দেন, তাহলে ইএক্সটিথ্রি চলে তার ডিফল্ট সেটিং-এ, মানে 'data=ordered', মানে শুরুতে খানিক মৃণাল-মাণিক অন্তে সুভাষ ঘাই দিয়ে কোনাকুনি বনসাইয়ে সাজানো টোনা-টুনির নিরাপদ সংসার।

৩.৩।। রাইজারএফএস

কারনেল ভার্শন ২.৪ থেকে আনুষ্ঠানিক ভাবে, অফিশিয়ালি পাওয়া গেল, কিন্তু তার কিছু আগে থেকেই রাইজারএফএস একটা কারনেল প্যাচের আকারে পাওয়া যেত। প্যাচ বলতে একদম তাপ্পি বলতে আমরা যা বুঝি। শুধু নেগেটিভ নয়, পজিটিভ তাপ্পি। মূল প্যান্টুলুনের ফুটো ঢাকতে তাপ্পি মারার কথা না-ভেবে ভাবুন একটা বাড়তি পকেট পাওয়ার আশায় পজিটিভ তাপ্পি। প্যাচকে মূল কাঠামোর গায়ে সেঁটে দেওয়া হয় বাড়তি কিছু পাওয়ার আশায়। বাড়তি কোনো কাজ, বাড়তি কোনো সুবিধা। ল্যারি ওয়াল আর পল এগার্টের এই প্যাচের সফটওয়ার ধারণাটা বেশ ইন্টারেস্টিং। কিন্তু এখানে সেটা আলোচনার জায়গা নয়। কারনেলের মধ্যে যা নেই, কিন্তু বাইরে থেকে যোগ করে নেওয়া যেতে পারে, কারনেলের ভার্শন স্থির হয়ে যাওয়ার আর যা কাজ হয়েছে সেগুলো দিয়ে দেওয়া যায় এই প্যাচে। একটু '`man patch`' করে নিতে পারেন। প্যাচ বুঝতে গেলে আবার একটু ডিফ (diff) বুঝতে হবে, ডিফ পড়ে দেখুন, দুটো ফাইলের তফাত বা ডিফারেন্স নিয়ে কাজ করে, খুব মজার।

রাইজার যাদের তৈরি সেই কোম্পানির নাম নেমসিস-কে (Namesys) খুব সহজেই নেমেসিস পড়া যায়, কিন্তু গ্রীক পুরাণ মাফিক তথ্যের কোনো সমুহ বিনাশ আদৌ এর উদ্দেশ্য নয়, বরং তার উল্টোটা, রাইজারএফএস এসেছিল পুরোনো ইএক্সটিটু-র একটা শক্তিশালী বিকল্প হিশেবে, তথ্যের শক্তিশালীতর যত্নআত্তির আশায়। সীমিত ডিস্কভূমির আরো কাবেল ব্যবহার, আরো কম জায়গায় মোট আরো তথ্য রাখা, রাইজারের একটা জোরের জায়গা। এবং ক্র্যাশ বা কেলো থেকে দ্রুততর আরোগ্য, যার কথা আমরা আগেই বলেছি। কিন্তু একটা সমস্যার জায়গা এই যে রাইজারের বেশি মনোযোগ ডেটার চেয়েও মেটাডেটার দিকে। পরে হয়ত রাইজার শুধু মেটাডেটা নয়, ডেটারও জার্নাল রাখতে শুরু করবে, ইএক্সটিথ্রির বেলায় যেমন বলছিলাম।

ডিস্কভূমির খুব ভালো ব্যবহার হয় রাইজারে। এই কাজে রাইজারএফএস মূলত ব্যবহার করে ব্যালান্সড-ট্রি অ্যালগরিদম। শুধু রাইজারএফএস নয়, এই ব্যালান্সড ট্রি অ্যালগরিদম ব্যবহার করে এইচএফএস (HFS), এনএসএস (NSS) এবং স্পাইরালগ (Spiralog) ফাইলসিস্টেমও। যদিও রাইজারএফএস ভার্শন ফোর, ব্যালান্সড ট্রি-র জায়গায় ড্যান্সিং ট্রি অ্যালগরিদমের কথা বলেছে। কিন্তু এটা এমন একটা জায়গায় ঢুকে পড়ছি, আমি ঢোকার চেষ্টা করে দেখেছি, অন্তত এখনো আমার সেখানে কথা বলার যোগ্যতা নেই। আমি আপনাদের এই রাইজার, এরপর জেএফএস

আর এক্সএফএস-এর একটা হালকা উপরসা আন্দাজ দিয়ে বেরিয়ে যাব। যতটুকু কথা না-জানলে অন্য খুব সাধারণ কাজগুলোও বুঝতে অসুবিধে হয়। কাজ করতে গিয়ে আমি দেখেছি, রাইজারএফএস আর এক্সএফএস অনেক সুবিধাজনক, রাইজার ব্যবহারের পরামর্শ আমায় দিয়েছিল অরিজিৎ, আর এক্সএফএসেরটা দিয়েছিল তথাগত। এবং ইএক্সটিটু বা ইএক্সটিথ্রির সঙ্গে গতির পার্থক্য এতটাই যে একবার ম্যানড্রেক ৮.২-এ এক্সএফএসে অভ্যস্ত হয়ে যাওয়ার পরে রেডহ্যাটে ইএক্সটিটু বা থ্রিতে, বিশেষ করে কিছু কম্পাইল করার সময়, আমার কান্না পেয়ে যেত। রেডহ্যাট উড়িয়েই দিয়েছিলাম। শেষে ইনস্টল করার সময়ে দ্বিতীয় ভার্চুয়াল কনসোলে গিয়ে 'mkfs.reiserfs' ব্যবহার করে পার্টিশনটা রাইজার বানিয়ে এক নম্বর কনসোলে ফেরত এসে পার্টিশন আর ফরম্যাট না-করে ইনস্টল করে, রাইজারে রেডহ্যাট করেছিলাম, তবে সিস্টেম ভদ্রস্থ হয়েছিল। রেডহ্যাটে এক্সএফএস করার কোনো উপায় আমি পাইনি, তবে ইন্দ্র বলেছিল ওর কাছে কী একটা সিডি আছে যা দিয়ে এক্সএফএসে করা যায়, আমায় দেবে, তা ইন্দ্র যা ব্যস্ত থাকে, ফোন করবে এবং ফোন করল এই দুটোর মধ্যে থাকে একমাসের দূরত্ব, যে আমি সামনের জন্মের আগে ওটা পাওয়ার খুব বেশি আশা করিনা। আমার নিজের অন্তত এক্সএফএস সবচেয়ে পছন্দের, কিন্তু স্ল্যাকওয়্যার নাইনে আমায় এক্সএফএস থেকে রাইজারে ফেরত যেতে হয়েছে, কারণ এক্সএফএস কারনেল যেটা থাকে সেটার সঙ্গে কিছু কিছু কারনেল মডিউলের বেশ সমস্যা হয়েছে। হতে পারে আমার এনভিডিয়া জিফোর্স কার্ডের জন্যে। কারণ, জিফোর্সের ড্রাইভারে দেখেছি কিছু সমস্যা হয়। আমার স্ল্যাকওয়্যার সিস্টেমে এক্সএফএস কারনেলে, এমপ্লেয়ার আর জিএসএল মানে গ্নু সায়েন্টিফিক লাইব্রেরি, সি-তে গণিত আর বিজ্ঞানের কাজ করার একটা লাইব্রেরি, কিছুতেই কম্পাইলড হচ্ছিল না। রাইজারে ফেরত আসার পর আবার সব চমৎকার হয়ে গেছে। দেখি, স্ল্যাকওয়্যার নাইন পয়েন্ট ওয়ানের সিডি ইমেজ ডাউনলোড করছে তথাগত, কালই ফোন করে জানিয়েছে, অনেকটাই হয়ে গেছে, যদি সেটায় ভালো ভাবে এক্সএফএসে করা যায় তাহলে আমার সবচেয়ে পছন্দের ডিস্ট্রো স্ল্যাকওয়ারে সবচেয়ে পছন্দের ফাইলসিস্টেম এক্সএফএসে কাজ করতে পারব। একই সঙ্গে এটাও বলে রাখা ভাল এই পছন্দগুলো কোনো রকমেই সব জেনে বুঝে পছন্দ নয়। হতেই পারে অন্য কোনো কিছু আমার আরো পছন্দ হত একটু বিশেষ রকমে বদলে নিলেই, যাকে টেকনিকাল ভাষায় বলে 'টুইক করা'। সেটার কথা আদৌ জানিই-না, এত কম জানি কম্পিউটারের। তবে, চারদিকের অবস্থা যা, কালকেই যাদবপুরে অংকুর বাংলা সিডি নিয়ে গ্নু-লিনাক্স সেমিনারের গল্প শুনছিলাম। সেখানে একজন সিনিয়র সিটিজেন কোনো একটা মন্তব্যের সূত্রে বলেছেন, আমি তো 'লিনাক্স এইট' ব্যবহার করি। এবং এই ঘটনাটার উপাখ্যানতার মূল খোঁচটা এইখানে যে লোকটা যাদবপুর কম্পিউটার সায়েন্স ডিপার্টমেন্টের ফ্যাকাল্টি মেম্বার, সেখানে পড়ায়, এবং সে, নাকি, লিনাক্স নিয়ে কাজ করে। এটা যদি আমি বলতাম, আমাদের জিএলটি-র নতুন ছেলেদের কেউ বলত, সেটা অবাক লাগত, এমা কিছুই তো জানিস না — লিনাক্স এইট বলেই কিছু হয়না। কিন্তু এই পরিস্থিতিতে এটা একটা অপদার্থ ধৃষ্টতা, একটা স্পর্ধা, একটা জাতিসত্তার ধর্ষণ। ইংরেজরা আমাদের যেমন চাইত তেমনি হয়ে থাকা। চারপাশটাই এইরকমই — একটা শক্তির কেন্দ্র, তার চারদিকে ছোট ছোট বোকা বোকা জমিদাররা। সমস্ত বিশ্বাস মূল্যবোধ সবকিছুই, ঠিক ওই লোকটার লিনাক্সে কাজের মতই, একটা জনশ্রুতি কিম্বদন্তী হিয়ারসে — 'নাকি' কাজ করে। অ্যাকাডেমির ওই সিনিয়র সিটিজেনদের মহিমাময় হয়ে বিরাজ করার কল্যানেই আমার পাড়ার ছোট ছোট ছেলেদের মধ্যে কম্পিউটার নিয়ে মহিমার ত্রাসের এলিটতার আবহ বানানো হয়, বয়ে নিয়ে যাওয়া হয়। সবকিছুই নিজে করার, নিজে করে মিলিয়ে নেওয়ার উপায় থাকে বলেই, গ্নু-লিনাক্স আসলে এই আবহটার উপরেই আক্রমণ। এই গোটা লেখাটায় যা করে চলার চেষ্টা করছি, পরিশ্রম করো, শেখো, সবকিছু তোমার জন্যে দেওয়া আছে গ্নু-লিনাক্সে। এমনকি আমিও, কম্পিউটারের বাইরের লোক হয়ে, যদি নিজের চেষ্টায় দুবছরেরও কমে এই লেখাটা লেখার মত শিখে উঠতে পারি, তাহলে তুইও পারবি। এটা আমি রোজ অশেষকে পিউকে কেয়াকে বলছি। সম্পদটা সামনেই আছে, দ্যাখ, শেখ। ওই মৌরসি পাট্টার বাইরে, ওদের পাণ্ডিত্যের জনশ্রুতির বাইরে, নিজে হাঁট। নিজে দেখে শেখার চেষ্টা কর, যা যা তোর কাছে আসছে তার সবই এক একটা প্রচার জনশ্রুতি রূপকথা, মূল্যবোধ মনন অবস্থানও এখন একটা জনশ্রুতি, একটা 'নাকি', নিজে হাঁট — যদি তোর ডাক শুনে কেউ না আসে, তবে 'নাকি' একলা চলতে হয় ... (কার্টসি নচিকেতা)

হ্যাঁ, রাইজারএফএস। রাইজারএফএস-এ গোটা তথ্যটাই সজ্জিত থাকে একটা কাঠামোয় যার নাম 'বি-ব্যালান্সড ট্রি'। এই ট্রি বা গাছের আকারে সজ্জার কল্যাণেই খুব চমৎকার ভাবে ডিস্কভূমিটাকে ব্যবহার করতে পারে রাইজারএফএস।

খুব ছোট ছোট ফাইলগুলোকে সরাসরি ওই গাছের পাতায় পাতায় রাখতে রাখতে যায়, শিকড় গুঁড়ি ডালপালা এরকম কোথাও না-রেখে — শুধুমাত্র বাস্তব ডিস্কের বাস্তব শরীরে তার অবস্থানের পয়েন্টারটুকু লিখে রেখে। এই লিখে রাখাটাও এক কেবি বা চার কেবির এক একটা খণ্ডে ঘটেনা, ঘটে ঠিক যতটা আয়তন প্রয়োজন ততটা আয়তনের এক একটা অংশে। রাইজারএফএসের আর একটা ভালো জায়গা আইনোডগুলোকে গতিশীল বা ডায়নামিক রকমে ধার্য করার প্রক্রিয়া। ইএক্সটিটু জাতীয় পুরোনো ফাইলসিস্টেমগুলোর চেয়ে এটা ভালো, যেখানে ফাইলসিস্টেম বানানোর সময়েই স্থির করা থাকে আইনোডদের ঘনত্ব কতটা অব্দি হতে পারে। খুব ছোট ছোট ফাইলের বেলায়, রাইজারে, ফাইল আর ফাইলের 'stat_data' বা আইনোড তথ্য পাশাপাশি রাখা থাকে, যাতে মাত্র একবার ডিস্কে পড়ে বা লিখেই ফাইল সংক্রান্ত গোটা কাজটা সিস্টেম করে নিতে পারে। এতে ডিস্ক অনেক দ্রুত কাজ করে। আর জার্নালের কল্যাণে কোনো কেলো বা ক্র্যাশ থেকে ফেরত আসতে রাইজারের সময় লাগে মাত্র কয়েক সেকেন্ড। এমনকি গাবদা গাবদা সাইজের ফাইলসিস্টেমেও।

৩.৪ ।। জেএফএস

জার্নাল থাকাটা জেএফএস-এর নামেই ঘোষিত — জার্নালিং ফাইল সিস্টেম। এটা এনেছিল আইবিএম। গ্নু-লিনাক্সে জেএফএস এসেছে খুব অল্পদিন, দুহাজার সালে বিটা ভার্শন বা প্রস্তুতিকালীন সংস্করণ, আর ২০০১-এ এসে প্রথম পূর্ণাঙ্গ ভার্শন। মূলত খুব বেশি কাজের চাপে থাকা সার্ভার মেশিনগুলোর কথা ভেবেই জেএফএস তৈরি, কাজ কত দ্রুত করা যায় এটাই যেখানে প্রায় একমাত্র বিবেচ্য। জেএফএস পুরোপুরি একটা ৬৪-বিট ফাইলসিস্টেম, তাই খুব বড় বড় সাইজের ফাইল এবং পার্টিশন বানানো জেএফএসে কোনো সমস্যাই নয়। আমরা এই সেকশনের শেষে বড় ফাইলের আয়তনের কথায় আসছি। সার্ভার সিস্টেমের বেলায় জেএফএস খুবই কার্যকরী।

রাইজারের মত জেএফএস ব্যবস্থাতেও জার্নাল রাখা হয় শুধু মেটাডেটার। খুব বিশদ সর্বব্যাপী চেকিং-এর জায়গায় চেক করা হয় ফাইলসিস্টেমের সাম্প্রতিকতম বদলের কারণে মেটাডেটার শরীরে ঘটে যাওয়া পরিবর্তনগুলোকেই। এতে ক্র্যাশ থেকে প্রত্যাবর্তনের প্রচুর সময় বেঁচে যায়। তথ্য লেখা এবং পড়ার কাজ, এবং তার সঙ্গে সঙ্গে লগ বা নথীর বদলের কাজ একই সঙ্গে একাধিক জায়গায় ঘটতে থাকে জেএফএসে, এক একটা গুচ্ছে বা দলে বা গ্রুপে। আলাদা আলাদা ভাবে লেখার কাজের জায়গায় এই কনকারেন্ট কাজের প্রথা জেএফএসের দ্রুততার একটা বড় কারণ। জেএফএসে দু ধরনের ডিরেক্টরি কাঠামো একই সঙ্গে থাকে। ছোট ছোট ডিরেক্টরির জন্যে ডিরেক্টরির মধ্যেকার মালপত্তর সরাসরি আইনোডে রাখা হয়। আর বড় বড় ডিরেক্টরির বেলায় বি-ট্রি ব্যবহার করা হয়। এতে ডিরেক্টরি সংগঠন খুব ভালো ভাবে হতে পারে। এবং আইনোড ধার্য করার প্রক্রিয়াটা জেএফএস ব্যবস্থাতেও গতিশীল বা ডায়নামিক। আমরা আগেই বলেছি ইএক্সটিটু ফাইলব্যবস্থায় এই আইনোড ঘনত্বটা আগে থেকেই স্থির-করা। তাই, ফাইলসিস্টেমে কতগুলো ডিরেক্টরি থাকতে পারবে, বা ডিরেক্টরিতে কতগুলো ফাইল থাকতে পারবে তার কিছু বিধিনিষেধ এসে পড়ে। জেএফএস ব্যবস্থায় এইসব ঝামেলা নেই, এখানে ডিরেক্টরি বা ফাইলদের ভূমি বণ্টন করা হয় গতিশীল রকমে। আর যেই কোনো একটা ফাইলের বা ডিরেক্টরির কোনো একটা জমির প্রয়োজন ফুরিয়ে যায় অমনি সেই জমিটাকে ফের বণ্টনের জন্যে তুলে রাখে।

৩.৫ ।। এক্সএফএস

ইউনিক্সের বিভিন্ন ফ্লেভারের কথা বলতে গিয়ে আমরা আইরিক্সের (Irix) কথা উল্লেখ করেছি, চার এবং পাঁচ নম্বর দিনের আলোচনায়। এসজিআই বা সিলিকন গ্রাফিক্স এই এক্সএফএস ফাইলসিস্টেম প্রথম তৈরি করে এই আইরিক্স অপারেটিং সিস্টেমের জন্যে, ১৯৯০-এ। খুব দ্রুত কাজ করতে পারে এমন একটা ৬৪-বিট ফাইলসিস্টেম বানানোর উদ্দেশ্যে। কম্পিউটারের তথা মেশিনের কাছে সিস্টেমের চাহিদা শুধুই বাড়ছিল। বড় বড় ফাইল নিয়ে কাজ করার ব্যাপারে এক্সএফএস খুবই দড়। পরে এক্সএফএসকে গ্নু-লিনাক্সে পোর্ট করা হয়। তবে ঠিক রাইজারএফএস বা জেএফএস-এর মতই এক্সএফএস-ও বেশি মাথা ঘামায় ডেটার চেয়েও মেটাডেটা নিয়ে। উন্নত হার্ডওয়ারে এক্সএফএস বোধহয় অন্য ফাইলসিস্টেমের চেয়ে দ্রুততর কাজ করতে পারে।

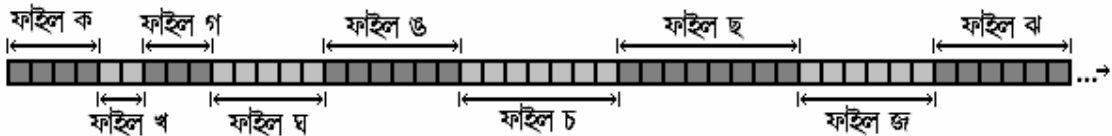
এক্সএফএস ফাইলসিস্টেম তৈরির সময়েই, যে ব্লক-ডিভাইসের উপর ফাইলসিস্টেমটা তৈরি হচ্ছে, সেটাকে আটটা বা তার বেশি সমান মাপের একরৈখিক বা লিনিয়ার অঞ্চলে ভেঙে ফেলা হয়। এদেরকে ডাকা হয় বণ্টন-দল বা

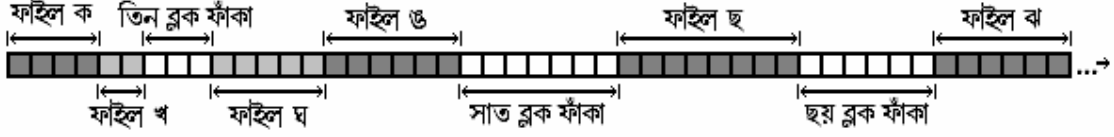
অ্যালোকেশন গ্রুপ বলে। প্রত্যেকটা অ্যালোকেশন গ্রুপের থাকে নিজের নিজের আইনোড এবং মুক্ত ডিস্ক ভূমি। এর মালিক ওই অ্যালোকেশন গ্রুপ। অ্যালোকেশন গ্রুপগুলো হল ফাইলসিস্টেমের মধ্যে ছোট ছোট ছানা ফাইলসিস্টেম। প্রত্যেকটা অ্যালোকেশন গ্রুপ অন্য অ্যালোকেশন গ্রুপদের থেকে স্বতন্ত্র এবং স্বাধীন। তাই কারনেল একই সাথে একাধিক অ্যালোকেশন গ্রুপের সঙ্গে কথোপকথনে আসতে পারে। জেএফএসের কনকারেন্ট ব্যবস্থার কথা মনে করুন। এই অ্যালোকেশন গ্রুপ হল এক্সএফএস ফাইলব্যবস্থার একটা জোরের জায়গা। কিন্তু, স্বাভাবিক ভাবেই, একাধিক প্রসেসর সম্পন্ন কম্পিউটারে, সিমেট্রিক-মাল্টি-প্রসেসিং গোছের, সবচেয়ে ভালোভাবে কাজ করতে পারে এই অ্যালোকেশন গ্রুপের ধারণাটা। এসএমপির কাজের সঙ্গে খুব যায় এই অ্যালোকেশন গ্রুপের ব্যবস্থাটা।
ডিস্কভূমিটাকে নিয়ন্ত্রণের কাজটা এক্সএফএস খুব ভালো ভাবে করে। এক্সএফএসের কাজের দ্রুততার একটা কারণ ওই বি-ট্রি। অন্য ফাইলসিস্টেম থেকে সম্পূর্ণ স্বতন্ত্র এক্সএফএস-এর একটা বৈশিষ্ট্য হল 'ঢিলে বণ্টন' বা ডিলেড অ্যালোকেশন ('delayed allocation')। ডিস্কভূমিতে তথ্য বণ্টনের গোটা কাজটাকে এক্সএফএস দুটো ভাগে ভাগ করে নেয়। স্থগিত রাখা বা দেরি-করিয়ে-দেওয়া বা ডিলেড কাজটাকে রেখে দেওয়া হয় র‍্যামে, আর তার জন্যে চিহ্নিত করে রাখা হয় প্রয়োজনীয় ভূমির পরিমাপ। কিন্তু সত্যিকারের বণ্টনটা করার জন্যে এক্সএফএস একদম শেষ মুহূর্ত অব্দি অপেক্ষা করে। এই ধরে রাখা তথ্যের মধ্যে কোনো অস্থায়ী অংশ যদি থাকে, অনেক তথ্যের চরিত্র তাই হয়, খুব ছোট্ট একটা সময়ের জন্যে জন্মায় তারা, তার পরেই মরে যায়, সেই তথ্য আর কোনোদিনই ডিস্কের গায়ে লেখা হয়না। কারণ, যতক্ষণে এক্সএফএস গোটা তথ্যটাকে লিখে ফেলার সিদ্ধান্ত নেয় তার মধ্যে হয়ত সে মরেই গেছে। একটা জিনিষ খেয়াল করুন, এই প্রক্রিয়ার কারণে মোট লেখালেখির পরিমাণ কিন্তু কমে যাচ্ছে। এটা এক্সএফএস ফাইলব্যবস্থার দ্রুততার একটা কারণ। আর ফাইলব্যবস্থার মধ্যে ফাইলগুলোর যে টুকরো হয়ে যাওয়া বা ফ্র্যাগমেন্টেশনের কথা বলেছিলাম আমরা, সেই ফ্র্যাগমেন্টেশনের পরিমাণও এতে কমে যাবে। অনেক কম সংখ্যক বার তাকে তো বাস্তব হার্ডডিস্কে লিখতে হচ্ছে। কিন্তু এই লেখালেখির প্রক্রিয়ার মাঝপথে সিস্টেমে যদি কোনো ঝাড় নামে, কোনো ক্র্যাশ, তাহলে অবভিয়াসলি তথ্য হারানোর পরিমাণও অন্য ফাইলসিস্টেমের চেয়ে বেশি হবে। এই টুকরো হয়ে যাওয়া বা ফ্র্যাগমেন্টেশন এড়ানোর আর একটা কায়দাও আছে এক্সএফএস-এর। যার নাম প্রাকবণ্টন বা প্রিঅ্যালোকেশন ('preallocation')। ফাইলটাকে ডিস্কের ফাইলসিস্টেমে লেখার আগে এক্সএফএস ফাইলটার সঙ্গে মানানসই কিছুটা ডিস্কভূমিকে রিজার্ভে সরিয়ে রাখে। তাই পরে যখন লেখে, স্বাভাবিকভাবেই, ফাইলটার টুকরো হওয়ার সম্ভাবনা অনেক কমে যায়। আর একটা ফাইলের মোট তথ্য যেহেতু একটা গোটা ডিস্কের নানা জায়গায় ছড়িয়ে থাকেনা, ফাইলসিস্টেমটা কাজও করতে পারে অনেক তাড়াতাড়ি।

৩.৬।। আইএসও-৯৬৬০ (iso9660)

সিডি পোড়ানোর প্রচুর সফটওয়ার গ্নু-লিনাক্সে এমনিতেই দেওয়া থাকে। ছবি-ইঁদুর দিয়েও রাশি রাশি সফটওয়ার আছে। ইন্দ্র জিএলটির মিটিং-এ একটা মজার কথা বলেছিল, এখানে প্রবলেমটা স্কেয়ারসিটি বা সফটওয়ারের অভাব নয়। প্রবলেম অফ প্লেনটি। প্রাচুর্যের সমস্যা। আসলে চারে এবং পাঁচে দেওয়া গ্নু-লিনাক্সের জন্ম এবং বিবর্তনের ইতিহাসটা খেয়াল করুন, তাহলেই বুঝতে পারবেন। ধরুন আপনি প্রোগ্রাম করতে জানেন, আপনি দেখছেন আপনার কোনো একটা কাজ করতে একটা সমস্যা হচ্ছে, আপনি চালু সফটওয়ারটাকে একটু বদলে নিতে চাইলেন। এই কাজের প্রোগ্রামের জন্যে যতগুলো সফটওয়ার পাওয়া যায়, তাদের সবাইকে আপনি নেড়ে ঘেঁটে দেখলেন। গ্নু-লিনাক্স জগতের বাইরে এটা অসম্ভব। কারণ, আপনি তো সোর্স কোড পাবেন না। আপনি প্রোগ্রামটা ব্যবহার করতে পারেন, জানতে বা বুঝতে পারেন না। এবার আপনার প্রয়োজন মোতাবেক সেটাকে মানিয়ে নিতে গিয়ে তৈরি হয়ে গেল একটা আদ্যোপান্ত নতুন প্রোগ্রাম। এবং স্টলম্যানের জিপিএলের খচড়ামিটা মনে করুন, আপনি জিপিএল-এর আওতায় আসা সফটওয়ার ব্যবহার করেছেন মানেই আপনার সফটওয়্যারও জিপিএলে এসে গেল, এটাও একইরকম উন্মুক্ত। এবং আপনি লিনাক্সী মানে আপনিও তো তাই চান। শুধু মাত্র নিজে পয়সা পিটে গাড়ি বাড়ি করলেন, আপনার নিজের পরবর্তী প্রজন্মও জানতে বুঝতে পারলনা, বিদ্যা থেমে গেল, এটা চাইলে তো আপনি গ্নু-লিনাক্সে আসতেনই না। এই প্রোগ্রাম বানাতে অন্য নানা প্রোগ্রামের চমৎকারিত্বটা গল্পের মত করে ভারি চমৎকার পাবেন এরিক রেমন্ডস-এর 'দি ক্যাথিড্রাল অ্যান্ড দি বাজার' লেখাটায় (http://www.tuxedo.org/~esr/)। এভাবেই গ্নু-লিনাক্সে যে কোনো কাজের জন্যে রাশি রাশি প্রোগ্রাম তৈরি হয়ে গেছে। ইন্দ্রও ঠিকই বলেছে, একজন নতুন ব্যবহারকারীর সত্যিই মাথা

খারাপ হয়ে যায়। তাও তো, এই লেখাটায়, আগেই ঠিক করে নিয়েছিলাম, কোনো গুই প্রোগ্রামকে আনব না। তাই সিডি পোড়ানোর যেসব কমান্ড মোড প্রোগ্রাম দেওয়া থাকে সিস্টেমে, তার মধ্যে দুটো হল সিডিআরডিএও (cdrdao) এবং সিডিরেকর্ড (cdrecord)। দুটোই বেড়ে জিনিষ। এই দুটোই হল, একটা সিডি রয়েছে, বা তার ইমেজ রয়েছে, সেটাকে একটা নতুন সিডিতে পোড়ানোর। কিন্তু ধরুন আপনার কিছু ডিরেক্টরি আর ফাইল আপনি সিডিতে তুলতে চান, সেই অবস্থায় আপনার সেই ডিরেক্টরি এবং ফাইলগুলো থেকে একটা ইমেজ বা প্রতিচিত্র বানাতে হবে। ইমেজটা আসলে কী করবে এখন আপনি সেটার একটা আন্দাজ করতে পারছেন। হার্ডডিস্কে থাকার সময়ে তার যে ফাইলব্যবস্থা ছিল সেটাকে বদলে সিডিতে থাকার ফাইলব্যবস্থায় নিয়ে আসা। এই ইমেজ বানানোর সফটওয়ারও দেওয়া থাকে গ্নু-লিনাক্সের ভিতরেই, কমান্ড মোডে করার, মেকআইএসওএফএস (mkisofs) এবং মেকহাইব্রিডএফএস (mkhybridfs)। নামদুটোকে দেখুন তো ভেঙে বুঝতে পারছেন কিনা। সফটওয়্যার দেওয়ার সঙ্গে সঙ্গে আরো চমৎকার যেটা, সেটা হল, এই প্রোগ্রামগুলোর নিজের ডকুমেন্টেশন, মূলত ম্যানপেজ পড়ে ফেললেই সিডি বিষয়ে বাংলা বাজারে নিজেকে একজন ধূরন্ধর বলে আপনি চালাতে পারবেন। ঠিক যেটা হয় এমপ্লেয়ারের (mplayer) ডকুমেন্টেশন পড়ে। বিভিন্ন ধরনের মাল্টিমিডিয়া ফাইল, তার ফরম্যাট, তার কাজের তফাৎ, একদম মাল্টিমিডিয়ার হদ্দমুদ্দ — যা বেবি, দৌড়ে যাবি, শিখবি মাল্টিমিডিয়া, নমস্তস্যৈ, নমস্তস্যৈ, নমস্তস্যৈ, নমোনমঃ। ম্যানপেজগুলো ব্রাউজ করে একবার দেখে নিন, আইএসও নিয়ে ওরা কী বলেছে।

সিডিরমের ফাইলব্যবস্থা নিয়ে একটু বড় করে এখানে বলতে হচ্ছে, তার কারণ, খেয়াল করুন, যে মাধ্যমটায় এই ফাইলব্যবস্থাটা তৈরি হচ্ছে সেটা একদম আলাদা। আমরা হার্ডডিস্কের চৌম্বক মাধ্যম নিয়ে অনেক কথা বলেছি। সিডির সেই অপ্টিকাল ভৌত মাধ্যমের আলোচনায় আর গেলাম না। শুধু মাধ্যম বদলে যাওয়ায় ফাইলব্যবস্থাটাও তো আমূল বদলে যাবেই, অন্যরকম হয়ে যাবেই। এবং মজার কথা কী বলুন তো, পুরোনো দিনের হার্ডডিস্কেও একসময় এই জাতীয় ফাইলব্যবস্থাই ব্যবহার হত, যা সিডিরমের বেলায় এখন হয়। আমরা আগেও প্রসঙ্গটা এনেছি, পরপর পরস্পর-সন্নিবদ্ধ বা কনটিগুয়াস ফাইলসিস্টেম। এই কনটিগুয়াস ফাইলসিস্টেম কিন্তু একটা জায়গায় ভারি উৎকৃষ্ট জিনিষ, ফাইল টুকরো হয়ে যাওয়ার বা ফ্র্যাগমেন্টেশনের কোনো সমস্যাই তাতে ছিলনা। কিন্তু কাজের ঝামেলার জন্যে সেটা বদলে যায়। পরে যা ফের ফেরত এল সিডি এবং ডিভিডি লেখার ব্যবস্থায়। কারণ এখানে ফাইল বা ডিরেক্টরিগুলো লেখার পরে আর বদলাচ্ছেনা। বারবার পোড়ানোর মত, মানে রিরাইটেবল সিডিতেও ভাবুন, আপনি যে পুড়িয়ে ফেলা ফাইল আর ডিরেক্টরিগুলোকে বদলাচ্ছেন, তা কিন্তু নয়। আপনি পুরোনো পোড়ানোটাকে মুছে ফেলে নতুন করে পোড়াচ্ছেন।

এই ছবিতে দেখুন, একটা কনটিগুয়াস ফাইলব্যবস্থায় ফাইল কী ভাবে থাকে। ফাইলব্যবস্থাটার শুরু থেকে শুরু করে মোট ৪৭-টা ব্লক আমরা দেখিয়েছি। পরপর একরৈখিক বা লিনিয়ার সজ্জায়, ফাইল ক থেকে ফাইল ঝ। ফাইলব্যবস্থাটা কিন্তু এর পরেও চলছে, এখানেই শেষ নয়, আমরা এখানে কয়েকটা মাত্র ফাইলকে দেখালাম। ফাইল ক থেকে ফাইল ঝ পরপর ৪, ২, ৩, ৫, ৬, ৭, ৮, ৬, এবং ৬ ব্লকের। আদতে এরা যে কোনো মাপেরই হতে পারত। ফাইলদের এই পরপর কনটিগুয়াস রকমে থাকাটা বেশ সহজ এবং আরামদায়ক একটা ব্যবস্থা। একটা ফাইলের মোট পরিমাপ মনে রাখা মানে জাস্ট দুটো সংখ্যা মনে রাখা, প্রথম ব্লকের নম্বর, আর ফাইলটায় কটা ব্লক আছে। যেমন ব্লকগুলোর নম্বর যদি ০ থেকে শুরু করি, কম্পিউটার-দুনিয়ায় মোটামুটি সেটাই প্রথা, আমাদের এই পাঠমালাটাও শুরু হয়েছিল ০ থেকে, তাহলে ফাইল ক মানে ০ আর ৪। শুরু ০ নম্বর ব্লক থেকে, চলেছে ৪ ব্লক। ফাইল খ মানে ৫ আর ২। ফাইল গ মানে ৭ আর ৩। ফাইল ঘ মানে ১০ আর ৫। ইত্যাদি। এই ফাইলব্যবস্থার আর একটা সুবিধে এই যে ফাইল পড়ার সময় সিস্টেমের সময় খরচ হবে সবচেয়ে কম, এক একটা ফাইল এক এক বারে গোটাটা পড়ে ফেলছে। একটা ফাইল পড়ার সময় সিস্টেমকে খুঁজতে হবে মাত্র একবার। শুধু শুরুর ব্লকটা গুনে বার করো, মার দিয়া কেল্লা। তার মানে এই কনটিগুয়াস ব্যবস্থাটা একই সঙ্গে সহজ এবং দ্রুত। তাহলে বাতিল হল কেন? গন্ডগোলটা পাকতে থাকে সময়ের সঙ্গে সঙ্গে। আপনি যদি আপনার মেশিনটাকে সচন্দন সবেলপাতা তুলে রাখেন, যেমন

অনেকেরই মেশিন দেখি, ঢাকনা লাগানো থাকে, যেন ওটা কাজের যন্ত্রপাতি নয়, একটা সযত্নে তুলে রাখার গয়না, তাহলে কোনো ঘাপলা নেই। আপনার ফাইলব্যবস্থাটা চিরকাল হুবহু একইরকম চমৎকারী আর মনোহর রয়ে যাবে। কিন্তু যদি আপনি ভুলক্রমেও মেশিনটা ব্যবহার করেন, তাহলেই গেলেন। পরের ছবিটা দেখুন।

আপনি মেশিনটা ব্যবহার করছেন। ব্যবহার করা মানেই কিছু পুরোনো ফাইল ওড়ানো, কিছু নতুন ফাইল লেখা। ফাইল গ, ফাইল চ এবং ফাইল জ ইহজগতে আর নেই, তাদের আত্মার শান্তিকামনা করে আপনি আরো আরো ফাইল লিখে জীবনে কিছু ফাইল রেখে যাওয়ার দিকে এগোচ্ছেন। এই ফাইল গ চ এবং জ তাদের ডিস্কপাতায় চৌম্বকজলের মত অস্থায়ী জীবন দিয়ে কিছুই যে রেখে যায়নি তা কিন্তু নয়। তারা প্রত্যেকেই ডিস্কের পাতায় এক একটা ফুটো রেখে গেছে, ঠিক যত ব্লকের উপস্থিতি ছিল তাদের, সেই মাপের এক একটা ফুটো। ডিস্কের প্রথম সাতচল্লিশটা ব্লক জুড়ে এখন আছে ছটা ফাইল এবং তিনটে ফুটো। ফুটো তিনটের সাইজ ৩, ৭ এবং ৬ ব্লক। এখনো আপনার সমূহ বেদনার দিন শুরু হয়নি। ফাইলব্যবস্থার আরো পরের আরো পরের দিকে, মানে আমাদের ছবির অদৃশ্য ডানদিকে আপনি ফাইল লিখে এবং লিখে চলেছেন। তাদের কিছু কিছু ওড়াচ্ছেনও নিজের খুশিমত। সেই বাবু আর কোথায় যারা পায়রা ওড়াত এবং বাজি পোড়াত, আমরা শুধু ফাইল ওড়াই এবং সিডি পোড়াই। একসময় আসবে যখন এইভাবে হরেক সাইজের ফুটোয় আর ফাইলে আপনার গোটা ডিস্কটা ভরে যাবে। এবার? এখন সিস্টেমকে ডিস্কে লেখার মত জায়গার হিশেব রাখতে হচ্ছে ফুটো এবং ফুটোর সাইজ গুনে, যাতে সেই ফুটোগুলোকে ঠিক ঠিক সাইজের ফাইলে ভরানো যায়। কোনো ফাইল লেখার আগেই এখন খুঁজে বার করতে হবে সেই ফাইল রাখার জন্যে সঠিক সাইজের একটা ফুটো। আপনি মেজাজে কবিতা লিখতে বসলেন, একটা ওয়ার্ড প্রসেসর খুললেন, প্রথমেই সিস্টেম আপনার কাছে জানতে চাইল, কত সাইজের ফাইল হবে? কারণ, সিস্টেমের তো মাথায় আছে ফুটোময় এই মহাবিশ্ব। আপনি হয়তো, আপনার প্রগাঢ় আত্মকাব্যসচেতনতার বলে বলীয়ান হয়ে, এবং পরে যাতে কাব্যোচ্ছ্বাস বাড়লে ফাইলসাইজের কার্পণ্যে আপনাকে আটকে যেতে না-হয়, কারণ, তখনতো ওই ফুটোয় আর তথ্য রাখা যাবেনা, আপনি জানিয়ে দিলেন, পাঁচ এমবি। সিস্টেম জানাল, বাতিল। ওই সাইজের ফুটো নেই। মাঝখান থেকে আপনার কবিতা লেখার মুডটাই ভেস্তে গেল। আবার ওয়ার্ড প্রসেসর খুলে চার এমবি তিন এমবি করে নিজের উচ্চাকাঙ্খা যে কমিয়ে আনতে পারেন না, তা নয়, কিন্তু, নিজেই ভেবে দেখুন, কবিতা লেখার তথা পিসি ব্যবহারের এটা একটা স্ট্যান্ডার্ড প্রক্রিয়া হল?

কিন্তু এবার ভাবুন, হার্ডডিস্কে যে ব্যবস্থা অসম্ভব, সেটা চমৎকার ভাবে সম্ভব হয়ে ওঠে যখন ফাইল বা ডিরেক্টরির কাঠামো এবং সাইজ এবং সংখ্যা আর বদলাবে না। ঠিক সিডি-রম এবং ডিভিডি-রমের বেলায় যেটা ঘটে। অনেক বছর আগে চৌম্বক হার্ডডিস্কে এই কনটিগুয়াস রকমেই ফাইল রাখা হত, এর সরলতা এবং দ্রুততার জন্যে। কিন্তু সিডি-রম, ডিভিডি-রম এই জাতীয় অপ্টিকাল মাধ্যমে আবার ফেরত এসেছে, যেখানে ফাইলব্যবস্থা মানে একবার লেখো এবং বারবার পড়ো। আইএসও-৯৬৬০ হল সিডি-রমের ফাইলব্যবস্থা। রম মানে মনে আছে তো, রিড-অনলি, শুধু পড়া যায়। এই রিড-অনলি ফাইলসিস্টেমের একটা আরাম ভাবুন, ফ্রি-ব্লক কী রইল তা নিয়ে মাথা ঘামাতে হয়না, আর তো লিখতে হবেনা। রিরাইটেবল সিডি মানে যেখানে পুরোনো তথ্য মুছে নতুন করে লেখা যায়, বা মাল্টিসেসন সিডি, মানে যেখানে আপনি সিডির গোটা তথ্যটা এক বারের জায়গায় একাধিক বারে পোড়াচ্ছেন, সেখানেও, লেখার মুহূর্তে গোটা ব্যবস্থাটা কিন্তু প্রদত্ত, পড়ার সময়ে তারা আর বদলাচ্ছে না। 'ইন-সিডি' জাতীয় যে ব্যবস্থাগুলো আছে, সিডিকেই ফ্লপির মত করে ব্যবহার করার, তাদের ফাইলব্যবস্থাগুলো, স্বাভাবিক ভাবেই, আলাদা।

আইএসও-৯৬৬০ হল সবচেয়ে জনপ্রিয় এবং বহুল ব্যবহারের সিডি-রম ফাইলব্যবস্থা। আইএসও-৯৬৬০ হল একটা আন্তর্জাতিক স্ট্যান্ডার্ড, অষ্টাশিতে গৃহীত হয়েছিল। বাজারে প্রাপ্তব্য প্রতিটি সিডি-রমই প্রাথমিকভাবে এই স্ট্যান্ডার্ড মেনে চলে, তার সঙ্গে কোথাও কোথাও আরো বাড়তি কিছু থাকে। এই স্ট্যান্ডার্ড গৃহীত হয়েছিল যাতে এই গ্রহের প্রতিটি সিডি-ড্রাইভেই যে কোনো সিডি-রম পড়া যায়, যে অপারেটিং সিস্টেমেই সেটা লাগানো হোক, যে রকম বাইট-ব্যবস্থাই ব্যবহার করা হোক সেই সিস্টেমে। পোসিক্স বা এলএসবি স্ট্যান্ডার্ডের ঠিক যে যুক্তি, আগেই বলেছি।

হার্ডডিস্কের বেলায় আমরা যে এককেন্দ্রিক সিলিন্ডারের কথা বলেছি, সিডি-রমে ওরকম কিছু থাকে না। তার বদলে থাকে একটা ছেদহীন স্পাইরাল বরাবর পরপর সাজানো বিটের সারি। একরৈখিক বা লিনিয়ার। তবে খোঁজার সময় স্পাইরালটার শুরু থেকেই শুরু করতে হবে, তা নয়। যে কোনো জায়গাতেই খোঁজা যেতে পারে। স্পাইরাল শব্দটার কোনো বাংলা প্রতিশব্দ আছে বলে জানিনা। একটা ছবি দিই। একটা বৃত্তাকার পথ, যার পরিধি ক্রমে কমছে বা বাড়ছে, যে ভাবেই ভাবুন। আপনাদের জন্যে আমি একটা স্পাইরাল আঁকার চেষ্টা করলাম, দু-ঘন্টা ধরে। পারলাম না। এটা শেষে নেটে গিয়ে নামিয়ে একটু বদলে নিলাম কপিরাইট বাঁচাতে। কিন্তু এরাই বা আঁকল কী করে? যন্ত্র দিয়ে মনে হয়। বা হাতে এঁকে স্ক্যান করে? কে জানে বাপু? প্রকৃতি ভীষণ স্পাইরালপরায়ণ। একটা শাঁখের মুখটা ভাবুন। স্পাইরাল আর ফিবোনাক্কি সংখ্যার সিরিজ নিয়ে ভারি মজা আছে। ফিবোনাক্কি সিরিজ মানে ০, ১, ১, ২, ৩, ৫, ৮, ১৩, ২১ ... এই সংখ্যার সারিটা, যার প্রত্যেকটাকেই পাওয়া যায় আগের দুটোকে যোগ করে। টিভিতে একদিন দেখাচ্ছিল, প্রকৃতির সব স্পাইরালগুলো এই ফিবোনাক্কি সিরিজ মেনে চলে।

যাকগে, সিডিরমের কথায় আসা যাক। সিডিরমের এই স্পাইরাল বরাবর বাইটগুলো পরপর সাজানো থাকে লজিকাল ব্লকের এককে। যার এক একটার মাপ ২৩৫২ বাইট। এর মধ্যে অনেকটা থাকে প্রিঅ্যাম্বল মানে গৌরচন্দ্রিকা — ধরুন মেটাডেটা গোছের, কী তথ্য কী ভাবে আছে সেই বিষয়ে তথ্য। এর সঙ্গে এরর কারেকশন বা ত্রুটি সংশোধনও থাকে। মানে তথ্যের আকারটা সঠিক আছে কিনা সেটা মিলিয়ে নেওয়ার কৌশল। আর জেনুইন তথ্য থাকে ২০৪৮ বাইট। গানের অডিয়ো সিডি থাকলে গানের আগের, পরের, মধ্যের নৈঃশব্দ্য। যা ডেটা সিডি বা তথ্যের সিডিতে থাকেনা। সিডি পোড়ানোর সময় দেখবেন, অনেক সময় মিনিট আর সেকেন্ডের মাপে গোটাটা বলছে, এই মাপটা হল ১ সেকেন্ড = ৭৫ ব্লক। প্রত্যেকটা সিডিরমের গোড়ায় থাকে ফাঁকা ষোলোটা ব্লক। ফাঁকা কারণ এগুলো নিয়ে আইএসও স্ট্যান্ডার্ডে কিছু বলে দেওয়া হয়নি। বুটেবল সিডি বানানোর জন্যে, মানে যা দিয়ে বুট করা যায়, অপারেটিং সিস্টেম বা অপারেটিং সিস্টেম চালু করে দেওয়ার যন্ত্রপাতি যাতে দেওয়া থাকে — এই বুট সিডি বানানোর জন্যে এই ষোলোটা ব্লক ব্যবহার করা হয়। অন্য কাজেও কেউ ব্যবহার করতে পারে। এরপর আসে প্রাইমারি ভলিউম ডেসক্রিপটর। এখানে একটা সিডিকে একটা সেটের একটা ভলিউম হিশেবে ভাবুন। এবার এই প্রাইমারি ভলিউম ডেসক্রিপটরে থাকে ওই সেটটার পরিচিতি, ভলিউমটার পরিচিতি, প্রকাশক এবং তথ্যটা যে সিডির জন্যে প্রস্তুত করেছে, তার পরিচিতি। সব সিস্টেমেই সিডিটা যাতে সমানভাবে পাঠযোগ্য হয়, তার জন্যে প্রাইমারি ভলিউম ডেসক্রিপটরের বিবরণে ব্যবহার করতে হবে শুধু বড় হাতের বর্ণমালা, সংখ্যা, আর অল্প কয়েকটা নির্দিষ্ট যতিচিহ্ন।

সিডিরমটার রুট ডিরেক্টরি এবং অন্য ডিরেক্টরিগুলো ভরা থাকে তাদের মধ্যেকার সাবডিরেক্টরি এবং ফাইলগুলোর এন্ট্রি দিয়ে। খেয়াল রাখুন, এটাও সেই অর্থে মেটাডেটা। এর শেষ ডিরেক্টরির শেষ এন্ট্রির মাথায় থাকে একটা বিশেষ বিট যা দিয়ে বোঝা যায় যে এখানে এই ডিরেক্টরি এন্ট্রিদের বিবরণ শেষ হল। ডিরেক্টরি এন্ট্রিগুলোয় থাকে ফাইলের ঠিকানা, ফাইলের সাইজ, ফাইলের তারিখ আর সময়, ফাইলের নাম। ফাইলের ঠিকানা মানে তার শুরুর ব্লকের নম্বর। ফাইলের নামের একটা কাঠামো আছে। মূল নাম বা বেসনেম, তারপরে একটা '.', তারপরে এক্সটেনশন, তারপরে একটা সেমিকোলন বা '; ', এবং সবার শেষে একটা বাইনারি ভার্শন নাম্বার। বেসনেম এবং এক্সটেনশন অংশে ছোটহাতের বা বড়হাতের বর্ণমালা এবং শূন্য থেকে নয় অব্দি অঙ্ক ব্যবহার করা যাবে। মূল নাম বা বেসনেমে আটটা অব্দি বর্ণ ব্যবহার করা যাবে, এবং এক্সটেনশনে তিন অব্দি। আপনার কি একটুও চেনা লাগছে এই হিশেবটা? ছয় নম্বর দিনের আলোচনার সঙ্গে মিলিয়ে?

এই তারিখ আর সময়ের ব্যাপারটায় একটু মজা আছে। আমরা Y2K বা ২০০০ সালে কম্পিউটার সিস্টেমের সময় এবং হিশেব রাখার সমস্যা নিয়ে বহ্বারম্ভে লঘুক্রিয়ার গল্প সবাই জানি। ঠিক এরকমই একটা সমস্যা আছে সিডিতেও। এই সময়ের ফিল্ডটায় একটা আলাদা বাইট করে দেওয়া থাকে বছর, মাস, দিন, ঘন্টা, মিনিট, সেকেন্ড, আর টাইমজোনের প্রত্যেকটার জন্যে। টাইমজোন বা সময়এলাকা ভৌগোলিক অবস্থান, গ্রিনউইচের তুলনায়, যেভাবে আমরা একটা দেশের সময় থেকে আর একটা দেশের সময়কে আলাদা করি। এবার, মজাটা ওই বছরের বাইট-টাতে। সময় গোনা শুরু হয়েছিল, ওই আইএসও কমিটির সিদ্ধান্ত মোতাবেক ১৯০০ থেকে। এরপর ২৫৬ বছর অব্দি, মানে ২১৫৫ সাল অব্দি সিডিতে বছরসংখ্যা লেখা যাবে। তার পরে আবার এটা ফিরে যাবে ১৯০০-তে। তবে কোনো ভয়

নেই, ২১৫৫ অব্দি কেন ২০০৫-অব্দিই সিডি থাকে কিনা সন্দেহ। পশ্চিমে সিডি এখন বাতিল প্রকৌশলের দলে। এখানেও, আমাদের লাগেরই, যে কজনেরই একটু পয়সা আছে, সব ডিভিডি এবং ডিভিডি ড্রাইভ ব্যবহার করছে। ডিভিডি বার্নারের এখনো অবশ্য অশ্লীল দাম। আচ্ছা, বলুন তো, কেন ২১৫৫ সালের পরে গিয়েই সমস্যাটা হচ্ছে? যদি ভেবে উঠতে না-পারেন, আপনার সময় এসে গেছে শূন্য থেকে শুরু করার, মানে, ফের শূন্য নম্বর দিন থেকে একবার পড়তে পড়তে আসুন।

ডিরেক্টরি এন্ট্রিগুলো রাখা থাকে বর্ণানুক্রমে, পরপর অক্ষর অনুযায়ী। শুধু প্রথম দুটো এন্ট্রি বাদ দিয়ে। প্রথমটা হল ডিরেক্টরিটা নিজেই। ঠিক গ্নু-লিনাক্সে যেমন, '.' মানে সেই ডিরেক্টরিটা নিজেই, আর '..' মানে জনিতৃ ডিরেক্টরিটা, মানে, যে ডিরেক্টরির সাবডিরেক্টরি এই ডিরেক্টরিটা। ডিরেক্টরি এন্ট্রিগুলোর প্রথম দুটো সবসময়েই থাকে এই দুটো এন্ট্রি। একটা ডিরেক্টরিতে কতগুলো এন্ট্রি থাকতে পারবে তার কোনো নিষেধ নেই, কিন্তু কটা স্তর অব্দি নামা যাবে, তার নিয়ম আছে। এই ডিরেক্টরির মধ্যে সাবডিরেক্টরি, মা থেকে ছানা, এই ভাবে নামা যাবে আটটা স্তর অব্দি, একে বলে নেস্টিং (nesting)। ডিরেক্টরি এন্ট্রির একদম শেষে এসে আর একটা জিনিষ থাকতে পারে, তার নাম 'সিস্টেম ইউজ' (System Use)। এই ফিল্ডটা সবসময় সব সিস্টেমে থাকেনা, এবং অনেক সিস্টেম এদের আলাদা আলাদা রকমে ব্যবহার করে। এই নেস্টিং আর ফাইলনামের কাঠামোর বিধিনিষেধ আর 'সিস্টেম-ইউজ' নামে ফিল্ডটার কথায় আসছি একটু বাদেই। সিডিরমের শুরুতেই এই ডিরেক্টরি এন্ট্রিগুলো শেষ হলে শুরু হয় ফাইল। ফাইলগুলো রাখা থাকে ওই পরপর কনটিগুয়াস ব্লকে। যেমন আগেই বলেছি, ডিরেক্টরি এন্ট্রিতে দেওয়া ফাইলের শুরুর ব্লকের ঠিকানা আর ফাইলের সাইজ দিয়ে একটা ফাইলকে পুরোপুরি বুঝে নেওয়া যায়। খুব ছোট করে, খুব সরল করে এই হল আইএসও ৯৬৬০ ফাইলব্যবস্থার কাঠামো।

> শুনুন, একদম বাইরের একটা মজা আপনাদের না-শুনিয়ে পারছিনা, এইমাত্র, এই শৈত্যতরঙ্গাক্রান্ত সকালে, একটা অটো ঢুকল আমাদের পাড়ায়। প্রত্যেকটা মফস্বলের কিছু নিজস্ব ভাঁড়ামি থাকে, মহানগরের ভাঁড়ামি থেকে স্বতন্ত্র — সেরকমই একটা, পুষ্পপ্রদর্শনী, মধ্যমগ্রামে হর বছর হয়। ওসব ডেসিবেল ছাড়ুন, মাইক নিয়ে প্রবল চেঁচাচ্ছে। একবার ভাবলাম বাইরে গিয়ে জিগেশ করি, পুষ্পকে যে প্রদর্শন করবেন, পুষ্পর বরের কোনো আপত্তি নেই তো? এর মধ্যেই এল ক্লাইম্যাক্সটা। এরা চেঁচানো ছেড়ে গান বাজানো শুরু করল, অটোটা পাড়া পেরিয়ে যেতে যেতে গানটার বেশ কয়েকলাইন শোনা গেল। কী গান হতে পারে, একবার মনে মনে কল্পনা করুন, যদি পারেন, এইমাত্র শীতের সকালে আমায় যে গরম মাছভাজাটা দেওয়া হল, কাল আমার জলের ফিল্টার পরিস্কার করার বখশিশ হিশেবে, সেটা আপনাকে দিয়ে দেব। গানটা কী, পারলেন ভাবতে? দেবব্রতর 'পুষ্প দিয়ে মারো যারে —'। পাগলা মাছ খাবি তো মাটি খা।

আইএসও ৯৬৬০-র তিনটে পর্যায় বা লেভেল আছে। লেভেল ওয়ানে বিধিনিষেধ খুব বেশি। ফাইলনামের ওই ৮+৩ কাঠামোটা রাখতেই হবে, বর্ণমালার নিয়মও মানতে হবে, ফাইলদেরও একদম নিখুঁতভাবে পরপর সন্নিহিত বা কনটিগুয়াস হতে হবে। এছাড়া আরো আছে। ডিরেক্টরি নামের কোনো এক্সটেনশন থাকবে না, তাদের হতে হবে আট অক্ষরের মধ্যে। এটা এসেছিল, বুঝতে পারছেন, এমএস-ডস ফাইলসিস্টেমের সঙ্গে মিলিয়ে। দু নম্বর লেভেলে নামের দৈর্ঘের নিষেধটা একটু শিথিল। সেখানে নামের অক্ষরের সংখ্যা হতে পারে ৩১ অব্দি, কিন্তু সেই বর্ণগুলো আসবে নিয়ম মেনে। তিন নম্বর লেভেলে ফাইলগুলোকে আর নিখুঁত ভাবে কনটিগুয়াস না-হলেও চলে, একটা ফাইল একাধিক অংশে ভাগ করা থাকতে পারে, তার এক একটা অংশের ব্লকগুলো কনটিগুয়াস হলেই হবে। এই রকম আরো কিছু।

আইএসও ৯৬৬০ প্রথার এই বিধিনিষেধগুলোকে বদলাতে এল রকরিজ এক্সটেনশন (Rock Ridge Extension)। রকরিজ নামটা এসেছে জিন ওয়াইল্ডারের 'ব্লেজিং স্যাডলস' নামে একটা সিনেমার একটা শহরের নাম থেকে, কারণ, যে কমিটি এই প্রথাটা স্থির করে তাদের একজনের খুব পছন্দ ছিল সিনেমাটা। রকরিজ এক্সটেনশন দিয়ে ফাইলের মালিকানা এবং অনুমতির যে কাঠামোটার আলোচনা করেছি আমরা সেটা রাখা চলে একটা সিডিরমে। রাখা যায় মেজর এবং মাইনর ডিভাইস নম্বর আর সিম্বলিক লিংক। ফাইলনামের দৈর্ঘ এবং বর্ণমালা সংক্রান্ত কোনো বিধিনিষেধ থাকেনা। রকরিজ দিয়ে আরো কিছু করা যায় — তার মধ্যে সবচেয়ে জরুরি পুনঃস্থাপন বা 'রিলোকেশন' (relocation)। যে রিলোকেশন দিয়ে ওই আট স্তর অব্দি নেস্টিং-এর নিষেধটাকে এড়ানো যায়। একটা ডিরেক্টরিকে আপনি রিলোকেশন করলেন এবং তার মধ্যে আবার আটটা স্তর অব্দি নেমে গেলেন। এই ভাবে। যে সিস্টেম রকরিজ চেনেনা সে এই সিডিরমটাকে রকরিজ বাদ দিয়ে এমনি সিডিরমের মত করেই পড়ে। আপনি কি এই রকরিজ

এক্সটেনশন কারা এনেছে সেটা নিজে বুঝে নিতে পারছেন? আইএসও ৯৬৬০ আসার পরপরই ইউনিক্সের বিভিন্ন ঘরানার লোকেরা মিলে যখন রকরিজ এক্সটেনশন প্রথা চালু করে তখনো গ্নু-লিনাক্সের জন্ম হয়নি। একটা জিনিষ খেয়াল করুন, রকরিজ যে বিধিনিষেধগুলোকে ভাঙল তার মধ্যে নামের দৈর্ঘের ফ্যাচাংটা কিন্তু উইনডোজ ৯৫ এবং তার পরবর্তী উইনডোজ ব্যবস্থার বেলাতেও অসুবিধে। কারণ, সেখানে নামের দৈর্ঘের কোনো নিষেধ নেই। এইজন্যে এল রকরিজ ছাড়াও আরো একটা জিনিষ, তার নাম জলিয়েট এক্সটেনশন (Joliet Extension)। জলিয়েট প্রথায় লম্বা ফাইলনাম আনা গেল, বর্ণমালার বিধিনিষেধ প্রায় নেই হয়ে গেল, আটটার বেশি নেস্টিং করা গেল, এবং সবচেয়ে মজার কথা ডিরেক্টরি নামেরও এক্সটেনশন দেওয়া গেল। এই শেষটা খুব অদ্ভুত। কারণ, উইনডোজে ডিরেক্টরি নামের এক্সটেনশন ব্যবহার হয়না।

আপনার ম্যানপেজ পড়ার একটা টাস্ক দেওয়া যাক। হার্ডডিস্কে থাকা কোনো একটা ডিরেক্টরি আর ফাইলের সমাহারকে সিডিরমের ফাইলসিস্টেম বানানোর সময়, আগেই বলেছি, একটা উপায় হল 'mkisofs'। এবার, ধরুন আমাদের জিএলটির সিডির জিনিষপত্তরগুলো, কয়েকহাজার ওয়েবপেজ আর কয়েকশো পিডিএফ সহ, রাখা আছে আমার মেশিনের '/arkive' ডিরেক্টরির মধ্যে '/arkive/linux.books' নামে একটা সাবডিরেক্টরিতে। এই গোটা ডিরেক্টরিটাকে আমি একটা ইমেজ করতে চাইছি 'glt-iso' নামে। এই ইমেজটাকে বানানোর জন্যে এর জন্যে সচরাচর আমি কমান্ড দিই, '`mkisofs -DJU -joliet-long -o glt-iso /arkive/linux.books`'। আপনি ম্যানপেজ পড়ে দেখুন তো, এই বাড়তি অপশানগুলো আমি কেন দিই? একটা কথা বলে রাখি, নামের বা নেস্টিং-এর কোনো বিধিনিষেধ মানা হয়নি, এবং আমি চাই আমাদের জিএলটির সিডিটাকে গ্নু-লিনাক্স বা উইনডোজ যে কোনো অপারেটিং সিস্টেম থেকেই সমানভাবে পড়া যাক। এর পরে, ইমেজ বানানো যেই হয়ে গেল, সিডিতে পোড়াতে হয় ইমেজটা। তখন ব্যবহার করছি 'cdrecord'। এখানেই বা অপশানগুলোর মানে কী? শুধু একটা জিনিষ বলি, সিডিরেকর্ড ব্যবহার করার আগে রুট হয়ে নেবেন। আর আপনার ডিভাইস কোনটা জানার জন্যে আপনি ব্যবহার করবেন 'cdrecord scanbus', এটাও রুট হয়ে। এটা আপনি ম্যান থেকেই পেতে পারতেন, কিন্তু কী করব বলুন, আমার মনটা এত নরম। এবার বলুন তো, '`-V "glt-mad@ilug-cal.org"`' অপশানটা 'mkisofs'-এর কমান্ডটার মধ্যে যোগ করলে তার মানে কী হত?

৩.৭।। দুম্বো সাইজের ফাইল আর ফাইলব্যবস্থা

এই গেল ছটা ফাইলসিস্টেম বুড়ি ছুঁয়ে আসা। এর পরেও আরো বহু ফাইলসিস্টেম ব্যবহার করা যায় গ্নু-লিনাক্সে। কিন্তু সেগুলো অনেক বিশেষ ধরনের প্রয়োজন। এখানে উল্লেখ করার মানে পাচ্ছি না। আর যদি তেমন দরকার পড়ে আপনি তো কোলকাতা লাগ এর মেইলিং লিস্টে (ilug-cal@ilug-cal.org) লিখতেই পারেন। মনে করে দেখুন, '`cat /proc/filesystems`' করে আমরা যে ফাইলসিস্টেমগুলোর তালিকা আমরা পেয়েছিলাম, তার বেশ কয়েকটার আলোচনা হল, কিন্তু কতকগুলোর হয়নি। সেগুলোর বিষয়ে প্রাথমিক ধারণা আপনার সিস্টেমের ভিতরেই কোথায় পাওয়া যাবে বলুন তো?

আগেই বলেছিলাম এই সেকশনের শেষে আমরা আসব একটা ফাইলসিস্টেম কত বড় সাইজের অব্দি হতে পারে, আর একটা ফাইলসিস্টেমে কোনো একটা একক ফাইল কত বড় সাইজের অব্দি হতে পারে, তার হিশেবে। এই হিশেবটা কিন্তু রোজই বদলাচ্ছে। রোজই নতুন প্রকৌশল আসছে, পুরোনো প্রকৌশল বদলাচ্ছে, সফটওয়ার এবং হার্ডওয়ার দুটো স্তরেই। তবু, একটা আন্দাজ করা যাবে। বড় ফাইলব্যবস্থার এবং ফাইলের যে বিকট হিশেব এখানে এসেছে, সেই সাইজের কোনো কিছু আমি দেখা বা চেনা তো দূরের কথা, ভেবেও উঠতে পারিনা। এখানে আমি যে হিশেবটা দিচ্ছি সেটা সুজে সাইট থেকে পাওয়া, এবং তাও একদম নতুন না।

হিশেবটা দেওয়ার আগে, বিট বাইট এবং তাদের বৃহত্তর এককগুলো নিয়ে দু-একটা কথা বলে নিই। শূন্যয় বলেছি আমরা, পরেও এসেছে, কিন্তু তখন আমরা বিট বাইটের উপর কিলো মেগা গিগা নিয়েই খুশি ছিলাম, টেরা-পেটা-এক্সা-জিটা-ইয়োটার বিকটতায় পৌঁছইনি। আর, এখানে এই পরিমাপগুলোর স্ট্যান্ডার্ড নিয়েও কিছু কথা বলার আছে। আমরা একাধিকবার বলেছি, কম্পিউটারের তথ্যের পরিমাপটা হয় বাইনারি পদ্ধতিতে, কিন্তু মজার কথা হল, এর গুণিতক বা মাল্টিপলগুলোর জন্যে যে প্রাকশব্দ বা প্রিফিক্স আমরা ব্যবহার করি, যেমন, কিলো, সেটা কিন্তু মেট্রিক পদ্ধতির। কিলোবাইট মানে আমরা ধরছি ১০২৪ বাইট, অথচ, মেট্রিক ব্যবস্থায় কিলো হল ১০০০। কারণ মেট্রিক

পদ্ধতি হল ডেসিমাল বা দশমিক। এখানে, আত্মীয়তাটা এইরকম যে ডেসিমাল সংখ্যা ১০০০ বা ১০^৩-এর সবচেয়ে কাছে বাইনারি প্রথায় পৌঁছচ্ছি ১০২৪ বা ২^১০। ('^' দিয়ে আমরা পাওয়ার বা সূচক বুঝিয়েছি, ২^৩ মানে ২×২×২) তাই মেট্রিক কিলো শব্দটা বাইনারিতে আসছে যখন, ১০০০ থেকে বদলে ১০২৪ হয়ে যাচ্ছে। আবার কিলো বলতে দশমিক কিলোও চালু ছিল, তখন লেখা হত ১ কিলোবাইট (বাইনারি) মানে ১০২৪ বাইট, ১ কিলোবাইট (দশমিক) মানে ১০০০ বাইট। কিন্তু এতে ভজকটোটা বেড়েছে বই কমেনি। '৯৮ সালের ডিসেম্বরে আইইসি (IEC — **I**nternational-**E**lectrotechnical-**C**ommission) একটা নতুন আইইসি স্ট্যান্ডার্ড চালু করে। তাতে বলা হয়, মেট্রিক ওই প্রাকশব্দ বা প্রিফিক্স বদলে বাইনারি প্রক্রিয়ার গুণিতকগুলোর জন্যে নতুন কিছু প্রাকশব্দ ব্যবহার করতে হবে। এই নতুন প্রাকশব্দগুলো তৈরি হবে মেট্রিক প্রাকশব্দের প্রথম দুই বর্ণ আর 'বাইনারি' এই শব্দটার প্রথম দুই বর্ণ মিলিয়ে। যেমন কিলোবাইট শব্দটা এখন থেকে হবে কিবিবাইট। কিবি (Kibi) মানে, কিলো (Kilo) থেকে আসছে কি (Ki) আর বাইনারি (Binary) থেকে আসছে বি (bi)। মেগাবাইটের 'মেগা' বদলে ব্যবহার করতে হবে মেবিবাইট, গিগাবাইটের 'গিগা' বদলে ব্যবহার করতে হবে গিবিবাইট, ইত্যাদি। কিন্তু এখনো আইইসি ব্যবস্থা তেমন জনপ্রিয় হয়নি। আমরা তাই সাবেক বাইনারি রকমেই হিশেবটা দিচ্ছি।

এক বিট (bit) = ০ বা ১, একক লেখা হয় 'b' দিয়ে।
এক বাইট (byte) = ৮ বিট, একক লেখা হয় 'B' দিয়ে।
এক কিলোবাইট (Kilobyte) = ১০২৪ বাইট = ২^১০ বাইট, একক লেখা হয় 'KB' দিয়ে।
এক মেগাবাইট (Megabyte) = ১০২৪ কিলোবাইট = ২^২০ বাইট, একক লেখা হয় 'MB' দিয়ে।
এক গিগাবাইট (Gigabyte) = ১০২৪ মেগাবাইট = ২^৩০ বাইট, একক লেখা হয় 'GB' দিয়ে।
এক টেরাবাইট (Terabyte) = ১০২৪ গিগাবাইট ২^৪০ বাইট, একক লেখা হয় 'TB' দিয়ে।
এক পেটাবাইট (Petabyte) = ১০২৪ টেরাবাইট = ২^৫০ বাইট, একক লেখা হয় 'PB' দিয়ে।
এক এক্সাবাইট (Exabyte) = ১০২৪ পেটাবাইট = ২^৬০ বাইট, একক লেখা হয় 'EB' দিয়ে।
এক জিটাবাইট (Zettabyte) = ১০২৪ এক্সাবাইট = ২^৭০ বাইট, একক লেখা হয় 'ZB' দিয়ে।
এক ইয়োটাবাইট (Yottabyte) = ১০২৪ জিটাবাইট = ২^৮০ বাইট, একক লেখা হয় 'YB' দিয়ে।

এবার এতাবৎ মহতী এককের বলে বলীয়ান হয়ে আসুন আমরা সেইসব জগদ্দল ফাইলব্যবস্থা আর ফাইল নিয়ে নাড়াচাড়া করি, ভেজানো মুগ আর বাদাম রেখেছেন তো? একদম প্রথমে গ্নু-লিনাক্সে দুই জিবি অব্দি সাইজের ফাইল বানানোর সংস্থান ছিল। কিন্তু তখনো মাল্টিমিডিয়া সেভাবে সিনে আসেনি। আর খুব বড় বড় সাইজের ডেটাবেস বা তথ্যভাণ্ডার নিয়ে কাজ করতে হয়নি। পরপর বড় বড় সার্ভারে যখন গ্নু-লিনাক্স দিয়ে কাজ হতে শুরু করল, কারনেল আর সি লাইব্রেরিগুলোকেও সেইমত বদলে নেওয়া হল যাতে দুই জিবির চেয়ে বড় সাইজের ফাইলও বানানো এবং ব্যবহার করা যায়। এককথায় এর নাম এলএফএস (LFS — **L**arge-**F**ile-**S**upport)। এখন প্রায় সবগুলো ফাইলব্যবস্থাতেই এলএফএস চালু আছে, যাতে খুব উন্নত ধরনের কাজকর্ম কম্পিউটারে করা যায়।

ফাইলসিস্টেম	ফাইল-সাইজের সীমা	ফাইলসিস্টেম-সাইজের সীমা
ইএক্সটিটু বা থ্রি, ১ কেবি ব্লক	১৬ জিবি (GB)	২ টিবি (TB)
ইএক্সটিটু বা থ্রি, ৪ কেবি ব্লক	২ টিবি (TB)	১৬ টিবি (TB)
রাইজারএফএস ৩.৫	৪ জিবি (GB)	১৬ টিবি (TB)
এক্সএফএস	৮ ইবি (EB)	৮ ইবি (EB)
জেএফএস, চার কেবি ব্লক	৮ ইবি (EB)	৪ পিবি (PB)

এর বাইরে গ্নু-লিনাক্স কারনেলের কিছু আভ্যন্তরীন ফাইলসাইজ সীমা আছে। ৩২-বিট সিস্টেমে একটা ফাইল বা ব্লক ডিভাইসের সর্বোচ্চ সাইজ হতে পারে ২ টিবি (TB)। তবে লজিকাল এলভিএম (LVM — **L**ogical-**V**olume-**M**anager) ব্যবহার করে এর চেয়েও বড় ফাইলব্যবস্থা বানানো যায়। ৬৪-বিট সিস্টেমে এই সীমাটা হল ৮ ইবি (EB)। উপরে আমরা কয়েকটা মাত্র ফাইলসিস্টেমের উদাহরণ দিলাম, এরকম সব ফাইলসিস্টেমেই দেওয়া যায়। এসব আমাদের বাড়ির নিজের পিসিতে কার লাগে কে জানে? আমার মেশিনে সবচেয়ে বড় ফাইল বানিয়েছিলাম, টেন কমান্ডমেন্টস এর চারটে ভিসিডি থেকে একটা এভিআই (*.avi) ফাইল বানিয়েছিলাম, এমপ্লেয়ারে ক্যাট করে করা

যায়। তার সাইজ ছিল আড়াই জিবি মানে ২.৬ গিগাবাইট। এটাই আমার টাইটানমানসের চরম। তথাগত ডিভিডি এনকোড ডিকোড করে, ওর হয়ত চার পাঁচ জিবি অব্দি লাগতে পারে। এগুলো সব তাদের লাগে যাদের সঙ্গে কম্পিউটারের সম্পর্ক আমার মত অটেকনিকাল নয়।

৪।। তুমি যে আমার, মাউন্ট, তুমি যে আমার

'মাউন্ট করা' — এই কথাটার হুবহু সংজ্ঞাটা হল একটা পার্টিশনে থাকা একটা ফাইলসিস্টেম (তথ্য-কাঠামো তথা ফাইল রাখার ব্যবস্থা অর্থে, যেমন এক্সএফএস বা ইএক্সটিথ্রি, আবার ডিরেক্টরিব্যবস্থা অর্থেও) একটা ঐক্যবদ্ধ ডিরেক্টরি হায়েরার্কিন্যস্ত ফাইলসিস্টেমে যুক্ত হয়ে পড়া। ঐ পার্টিশনের ফাইলব্যবস্থার মধ্যেও একটা হায়েরার্কিন্যস্ত ডিরেক্টরি ব্যবস্থা আছে। পার্টিশনের ফাইলব্যবস্থাটুকুর একটা রুট ডিরেক্টরি আছে। সেই রুট ডিরেক্টরি থেকে শাখা-প্রশাখার মত বেরিয়ে এসেছে নানা ডিরেক্টরি সাবডিরেক্টরি এবং তাদের মধ্যে থাকা ফাইল, যেমন আমরা দেখিয়েছি সাত নম্বর দিনের ১১ নম্বর সেকশনের ছবিতে। ঐক্যবদ্ধ মূল ফাইলসিস্টেমের তেমনি আবার একটা রুট আছে, এবং সেখান থেকে বেরোনো শাখাপ্রশাখা। পার্টিশনের ওই ফাইলব্যবস্থার রুট ডিরেক্টরি এখন মূল রুট ডিরেক্টরির মধ্যের কোনো একটা ডিরেক্টরি বা সাবডিরেক্টরিতে যুক্ত হয়ে যায়।

আজকের আলোচনার ১ নম্বর সেকশনের দ্বিতীয় ছবিটা দেখুন। যখন আমার সিস্টেমে '/dev/hda6' পার্টিশনটা মাউন্ট করা রয়েছে '/mnt/slackware' ডিরেক্টরিতে, তখন আমি '/mnt/slackware' ডিরেক্টরিতে দাঁড়িয়ে '`ls`' কমান্ড দিলে আমি পাব '/mnt/slackware/bin', '/mnt/slackware/boot', '/mnt/slackware/dev', '/mnt/slackware/etc' ইত্যাদি ডিরেক্টরিগুলো। এখন ওই ছবিতে একদম নিচের ডানদিকের নলপথের স্টপককটা খোলা আছে তাই মাউন্টপয়েন্ট '/mnt/slackware' বেয়ে তথ্য চলাচল করছে। এই ডিরেক্টরিগুলো স্ল্যাকওয়ার পার্টিশনের রুট ডিরেক্টরির মধ্যেকার সাব ডিরেক্টরি, স্ল্যাকওয়ার অপারেটিং সিস্টেমে বুট করলে তাদের স্ল্যাকওয়ার সিস্টেমের রুট পার্টিশনের সাবডিরেক্টরি হিশেবেই দেখাত। তখন, স্ল্যাকওয়ার অপারেটিং সিস্টেমে, '/home/dd' ডিরেক্টরিতে কোনো ফাইল আমি লিখে থাকলে এখন সেটাকে পাব '/mnt/slackware/home/dd' ডিরেক্টরির মধ্যে একই চেহারায়।

কিন্তু মাউন্ট না-করা অবস্থায়, যখন ওই স্টপককটা বন্ধ আছে, পার্টিশনটা বিচ্ছিন্ন হয়ে গেছে মূল ঐক্যবদ্ধ ফাইলব্যবস্থা থেকে, তখন আমি ওই একই '/mnt/slackware' ডিরেক্টরিতে দাঁড়িয়ে একই '`ls`' কমান্ড দিলে দেখাবে কিছুই নেই। কারণ তখন ওই পার্টিশনের নিজস্ব ফাইলব্যবস্থা মূল ঐক্যবদ্ধ ডিরেক্টরিন্যস্ত ফাইলব্যবস্থায় যুক্ত নেই, মানে মাউন্ট করা নেই। এই মাউন্ট করা হবে কি হবে-না, কোথায় হবে, কখন হবে সেগুলো আপনার ইচ্ছে অনুযায়ী ঘটবে। আপনি যদি নিজের কোনো ইচ্ছে আলাদা করে না জানিয়ে থাকেন, তাহলে সেটা হবে সিস্টেমের ডিফল্ট বা চালু পথে।

ঠিক তেমনি যখন '/mnt/windows/c' আর '/mnt/windows/d' নলপথের স্টপকক দুটো খোলা আছে, মানে '/dev/hda1' আর '/dev/hda2' পার্টিশনদুটো মাউন্ট করা আছে, তখন '/mnt/windows/c' আর '/mnt/windows/d' ডিরেক্টরিদুটোয় পাব উইনডোজের সি-ড্রাইভ আর ডি-ড্রাইভের গোটা ফাইলব্যবস্থা। স্টপকক বন্ধ হলেই, আনমাউন্ট করা হলেই তারা হারিয়ে যাবে। তখন আমার '`ls`' কমান্ড '/mnt/windows/c' আর '/mnt/windows/d' অব্দি এসে থেমে যাবে। একই ঘটনা ঘটবে '/dev/hdb5' পার্টিশন মাউন্ট করা থাকলে '/mnt/arkive' ডিরেক্টরিতে। মানে মূল ঐকবদ্ধ ফাইলব্যবস্থা কোনো একটা পার্টিশনকে যে কোনো সময় 'তুমি যে আমার' করে নিতে পারে, মাউন্ট করে, জাহাজে ভাসমান নাবিকদের মত যে বন্দরে সেটা করছে, সেই বন্দরটা হল মাউন্ট-পয়েন্ট।

একটা পার্টিশনকে আপনি যে কোনো ডিরেক্টরিতে মাউন্ট করতে পারেন, 'mount' কমান্ড দিয়ে। আমার সিস্টেমে অটোমাউন্ট অফ করা আছে, আগেই বলেছি, এমনি এমনি সিস্টেম মাউন্ট হয়না। ধরুন, আমি আমার '/arkive/linux.books' ডিরেক্টরিতে কোনো একটা ফাইল পড়তে চাইছি, এখন আমি পাব কী করে? মাউন্ট তো করা নেই। তার মানে এখন '`ls`' মারলে '/arkive' ডিরেক্টরি তো শূন্য দেখাবে। এই অবস্থায় আমায় কমান্ড দিতে হয়, '`mount /dev/hdb5 /arkive`'। এই কমান্ডটা রুট হয়ে দিতে হয়, যদিনা অন্য কোনো বিশেষ ব্যবস্থা করা থাকে, '/etc/fstab' বলে একটা ফাইলে। আমারটায় সেটা করা আছে, সেই কথায় আসছি একটু বাদেই। এবার এই গোটা কমান্ডটার কাঠামোটা খেয়াল করুন — প্রথমে মাউন্ট কমান্ডটা, তারপর পার্টিশনের নাম, তারপর সেই ডিরেক্টরির নাম যেখানে মাউন্ট করা হবে। এটা মাথায় রাখবেন, মাউন্ট করা হয় একটা পার্টিশনকে, করা হয় একটা

ডিরেক্টরিতে। এখন থেকে ওই পার্টিশনের ফাইলদের ওই ডিরেক্টরিতে পাওয়া যাবে। আদতে গোটা কমান্ডের কাঠামোটা আর একটু বড়, 'mount -t <fstype> -o <options> <device> <dir>'। এর মধ্যে আমরা প্রথম দুটো অংশ বাদ দিয়েছি, '-t'-এর পরে '<fstype>' অংশে দিতে হয় ফাইলসিস্টেমের মানে তথ্যকাঠামোর নাম। যেমন আমরা জানি আমাদের '/dev/hdb5' পার্টিশনটায় ফাইলসিস্টেম হল রাইজারএফএস (reiserfs), এখানে সেই নামটা দিতে হত। সচরাচর এটা না-দিলেও চলে, সিস্টেম নিজেই বুঝে নেয়। '-o'-এর পরে '<options>' অংশে বিভিন্ন রকম অপশান দেওয়া যায়। যেমন, আমরা হয়ত চাইছি শুধু রিড-অনলি রকমে পার্টিশনটাকে মাউন্ট করতে, কোনো ফাইল লিখতে আমরা চাইনা, শুধু পড়তে চাই, তখন অপশান দিতে হত 'ro' মানে রিড-অনলি। কিছু বলে না-দিলে সিস্টেম মাউন্ট করে তার ডিফল্ট রকমে, মানে, তাতে ফাইল লেখা আর পড়া দুই-ই করা যায়। তার মানে, তখন কমান্ড দিতে হত 'mount -o ro /dev/hdb5 /arkive'। এখানে '<device>' তো আমরা জানিই, '/dev/hdb5' নামের পার্টিশন, আর '<dir>' মানে ডিরেক্টরি মানে মাউন্টপয়েন্ট হল '/arkive'। ম্যানপেজে দেখুন। এইভাবে আমরা মাউন্ট করে নিতে পারি যে কোনো পার্টিশনে, এমনকি একাধিক জায়গাতেও মাউন্ট করা যায় একই ডিভাইসকে। বা মাউন্ট করা যায় একটা ইমেজ ফাইলকেও। সেসব আপনি ধীরে ধীরে শিখে যাবেন, হাউটুতে দেখবেন সব দেখা আছে। এক এক করে এইরকম সবগুলো পার্টিশনেই মাউন্ট করা যায়। নিজেই করে দেখুন।

একটু এবার '/etc/fstab' ফাইলটার সঙ্গে পরিচিত হয়ে নিন। আমার মেশিনের সুজে সিস্টেমের '/etc/fstab' ফাইলটা তুলে দিলাম এখানে। খেয়াল করুন, আমার মেশিনে আরো একটা 'fstab' ফাইল আছে। কোন ফাইলটা মনে করতে পারছেন? '/mnt/slackware/etc/fstab'। কেন বলুন তো? সিস্টেমের কনফিগারেশন ফাইল একটা সিস্টেমে তো একটার বেশি থাকার কথা না, তাহলে? আসছি সেই কথায় আগে '/etc/fstab' ফাইলটা চিনুন।

```
#device     mountpoint         fs         options                dump  fsck
/dev/hdb3   /                  xfs        defaults               1     1
/dev/hdb1   /boot              xfs        defaults               1     2
/dev/hdb2   swap               swap       pri=42                 0     0
/dev/hda1   /mnt/windows/c     vfat       rw,noauto,noexec       0     3
/dev/hda5   /mnt/windows/d     vfat       rw,noauto,noexec       0     3
/dev/hda6   /mnt/slackware     reiserfs   rw,noauto,user,exec    1     3
/dev/hdb5   /arkive            reiserfs   rw,noauto,exec         1     3
/dev/cdrom  /media/cdrom       auto       ro,noauto,user,exec    0     0
/dev/cdwri  /media/cdwri       auto       ro,noauto,user,exec    0     0
/dev/fd0    /media/floppy      auto       noauto,user,sync       0     0
devpts      /dev/pts           devpts     mode=0620,gid=5        0     0
proc        /proc              proc       defaults               0     0
usbdevfs    /proc/bus/usb      usbdevfs   noauto                 0     0
```

এই ফাইলটা কিন্তু হুবহু আমার সুজে সিস্টেমের '/etc/fstab' ফাইলটা নয়। একটু একটু বদলাতে হয়েছে, যেমন একদম শুরুর যে লাইনটা, '#' দিয়ে শুরু, ওটা সুজেতে থাকেনা, অন্য কোনো সিস্টেমে দেখেছি, বোঝার সুবিধের জন্যে এখানে যোগ করে দিয়েছি। '#' চিহ্নটা আদতে পাঠক বদলানোর জন্যে। এইটা কোনো লাইনের গোড়াতেই থাকা মানে, কারনেলকে বলে দেওয়া, ওহে সরলমতি কারনেল, এই লাইনটা ফর অ্যাডাল্টস, তোমাকে এটা পড়তে হবেনা। এটা শুধু জ্যান্ত ব্যবহারকারীরা পড়বে, যাদের হাত পা ঠান্ডা-লাগা সংস্কৃতি ইনকাম-ট্যাক্স ইত্যাদি আছে। ফর ইয়োর আইজ ওনলি, ভো রিডার, আপনি যাতে বুঝতে পারেন, কোথায় কী আছে। একবার পড়ে দেখুন, শেষদুটো এন্ট্রি বাদ দিয়ে আর সবগুলোই আপনি বুঝতে পারছেন। তাও, একদম শেষেরটা, 'fsck', যেন চেনাচেনা লাগছে, কোথায় একটা দেখেছিলেন? আমি মনে করিয়ে দিচ্ছি, সাত নম্বর দিনের ১১ নম্বর সেকশনে — এত খারাপ মেমরি নিয়ে চলেন কী করে? হাতে একটু টাকা এলে সিস্টেম আপডেট করে নিন। চাঁদনির বাজারে স্মৃতি চিপ প্রতি কত একটা দরে পাওয়া যাচ্ছে। আর একদম শেষ তিন লাইন আপনি আপাতত গয়ায় পিণ্ডদান করে দিন, ইনস্টল করার সময়ে সিস্টেমই ওগুলো লিখেছে, এই পাঠমালাতে আসবে না ওদের আলোচনা, আর আপনার আপাতত কাজেও লাগবে না, যখন লাগবে নিজেই শিখে নিতে পারবেন। এইমাত্র যে 'mount' কমান্ডের আলোচনা করলাম আমরা তার '<options>' অংশগুলোকে এবার 'man mount' করে মিলিয়ে নিন। দেখুন, প্রত্যেকটাই আপনি হুবহু বুঝতে

পারছেন। শুধু 'mount' ম্যানপেজ পড়ে হবেনা কিন্তু 'man fstab' করে সেই ম্যানপেজটাও পড়ে নিতে হবে। একটু আগে 'mount /dev/hdb5 /arkive' কমান্ড দিয়ে আমরা যে মাউন্ট করছিলাম '/dev/hdb5' নাবিককে, '/arkive' বন্দরে, সেই কমান্ডটা আমি ছোট করে 'mount /dev/hdb5' বা 'mount /arkive' দিতে পারতাম, তখনো একই কাজ হত, কারণ মাউন্ট অন্য প্রয়োজনীয় তথ্যগুলোকে 'fstab' ফাইল থেকে নিজেই পড়ে নিত।

'/etc/fstab' ফাইলের প্রথম মানে 'device' স্তম্ভটা তো সহজেই বুঝতে পারছেন। পার্টিশন ডিভাইসগুলোর নাম। দ্বিতীয় 'mountpoint' হল সেই ডিরেক্টরি যেখানে মাউন্ট করা হবে। আজকের আলোচনার দুই নম্বর মানে বাড়ির প্ল্যানের মত দেখতে ছবিটার সঙ্গে মিলিয়ে নিন। এর প্রথম এন্ট্রিটা রুট ফাইলসিস্টেম মানে '/' হওয়াটাই প্রথা। তার পরে 'fs' মানে ফাইলসিস্টেম, তথ্য তথা ফাইল রাখার কাঠামো, ইএক্সটিটু থ্রি জেএফএস এক্সএফএস ফ্যাটথার্টিটু ইত্যাদি যাদের পিন্ডি আমরা তিন নম্বর সেকশনে চটকালাম। এই স্তম্ভে দেখুন উইনডোজ পার্টিশন দুটোর ফাইলসিস্টেমের নাম 'vfat'। এটা হল উইনডোজ ফ্যাটথার্টিটুর গ্নু-লিনাক্স বিকল্প। উইনজোজের ওই পার্টিশনগুলোকে গ্নু-লিনাক্স ওই সিস্টেমেই পড়ে। ফাইলের নিচের দিকে নয়, দশ এবং এগারো নম্বর লাইনে দেখুন, তিনটে ডিভাইস, '/dev/cdrom', '/dev/cdwri' আর '/dev/fd0'। প্রথমটা সিডি পড়ার ড্রাইভ, দ্বিতীয়টা সিডি পোড়ানোর ড্রাইভ, আর তিন নম্বরটা ফ্লপি। এই তিনটেই কিন্তু লিংক ফাইল। নিজে 'ls -al' করে দেখুন, আপনি তো জানেন, কী করে দেখতে হয়।

আপনার সিস্টেমে যদি সিডি পোড়ানোর ড্রাইভ না-থাকে, তাহলে '/dev/cdwri'-টা থাকবে না। বা এগুলো অন্য কোনো নামেও থাকতে পারে, যেমন সুজে নিজে নিজে একটা বিচ্ছিরি লম্বা নাম দেয়, '/dev/cdrecorder', লাইন ভেঙে যায়, ফাইলটা দেখতে বিচ্ছিরি লাগে। ওই নামটা বদলে নেওয়া নিজের। এবং এই '/dev/cdrecorder' নামটাও সুজে বা অন্য কোনো অপারেটিং সিস্টেমই নিজে নিজে দেয়না। পাঁচ নম্বর দিনের ৬ নম্বর সেকশনে আমরা '/etc/lilo.conf' ফাইলটা নিয়ে আলোচনা করেছিলাম। তাতে কিছু লাইন আমরা বাদ দিয়ে গেছিলাম, এরকম একটা লাইন দেখুন, সুজে আর স্ল্যাকওয়ার দুটো অংশেই একবার করে মোট দুবার আছে। 'append = "hdc=ide-scsi"'। সুজে লাইনটায় আরো একটা অংশ আছে 'splash=0'। ওটা সুজের নিজস্ব কায়দা, ভুলে যান, সুজে ব্যবহার করলেই লাগবে কেবল। বুট করার সময় থকথকে রোমান্টিক নীল রঙে পঞ্চশর ভস্ম গোটা স্ক্রিনে ছড়িয়ে দেওয়ার অভ্যেস আছে সুজের, ছবিতে ছবিতে ছয়লাপ করে দেওয়ার, সেইসব কায়দা বাজি বন্ধ করতে বলা আছে। আর 'append = "hdc=ide-scsi"' অংশটা দিয়ে সুজে আর স্ল্যাকওয়ার দুটো সিস্টেমকেই বলা আছে, '/dev/hdc' নামের যে ডিভাইসটা, যদিও সেটা আইডিই ডিভাইস, সেটাকে স্কাসি ডিভাইস হিশেবে পড়ো। এরই অন্য নাম স্কাসি-নকল বা স্কাসি-এমুলেশন। এটা করতে হয়, সিস্টেমে সিডি পোড়াতে চাইলে, আপনার সিডি-ড্রাইভটা যদি বার্নার-ড্রাইভ হয়। তখনই ওই '/dev/cdrecorder' লিংকটা সিস্টেমের ডিভাইস ফাইলে বানানো হয়, যা আপনার '/dev/hdc' ডিভাইসটাকে আঙুল দেখায়। এখন থেকে যখনই আপনি লিংকটায় কিছু করছেন, লিখছেন বা পড়ছেন, আসলে করছেন সেটা আপনার বার্নার ডিভাইসটাকে।

'/etc/fstab' ফাইলের '/dev/cdrom', '/dev/cdwri' আর '/dev/fd0' এই তিনটে ডিভাইসেই দেখুন ফাইলসিস্টেম করা আছে 'auto'। মানে আপনি সিস্টেমের বিচারবুদ্ধির উপর ছেড়ে দিচ্ছেন। রিমুভেবল মানে বার-করে-ফেলা-যায়-এমন ডিভাইসে, সিডি ফ্লপি ইত্যাদি, এটাই চালু প্রথা। ম্যানপেজ পড়ে দেখুন আপনি এখানে একটু আগে পড়া জলিয়েট রকরিজ ইত্যাদিও ব্যবহার করতে পারেন। এখন তো আপনার সিডিকাঠামো বিষয়ে গুচ্ছ ফান্ডা। এর পরে আসে 'options' স্তম্ভটা, এতে আমাদের একটাই কথা বলার আছে, কোষ্ঠকাঠিন্যের সমস্যায় খেজুর অত্যন্ত উপকারী, যদিও, অপর্যাপ্ত ব্যবহারের আগে একবার পোস্ট-পন্ড্রিয়াল চেক করে নেওয়া ভালো। 'dump' স্তম্ভটায় আসতে পারে এক বা শূন্য। ডাম্প হল একটা ইউটিলিটি সফটওয়ার বা ব্যবহারিক প্রোগ্রামের নাম। ব্যাক-আপ করার। এই স্তম্ভে এক থাকা মানে ডাম্প সেই পার্টিশনকে ডাম্প করবে, শূন্য থাকা মানে করবে না, যখন আপনি ডাম্প চালাবেন। ডাম্প প্রোগ্রামটা আপনি চালালে, তখন কোন পার্টিশনকে কী করতে হবে সেটা ডাম্প এই 'fstab' ফাইল থেকে পড়ে নেবে। শেষ স্তম্ভ হল 'fsck'। ফাইলসিস্টেমকে চেক করা হবে কিনা, তার হাল-হকিকত পরখ করে নেওয়া হবে কিনা, হলে, কার পরে কাকে চেক করা হবে সেইটা বলে দিচ্ছে এই স্তম্ভের মানগুলো। গ্নু-লিনাক্স বুট করার সময়, যে

ফাইলসিস্টেমগুলোয় এই স্তম্ভের মান শূন্য করা নেই, তাদের চেক করে, কোনো দুর্নীতি ঘটেছে কিনা, কোথাও কোনো তথ্যকাঠামো ঘেঁটে গেছে কিনা, কোরাপশন আছে কিনা ফাইলব্যবস্থায়। আর এই স্তম্ভের সংখ্যা মেনে পরপর সেই চেকটা করে। যেমন, সচরাচর রুট পার্টিশনের থাকে এক, তার মানে প্রথম চেক করবে এই রুট পার্টিশনকে। তারপর দুই, তারপর তিন, এইভাবে। আপনার মেশিনে যখন গ্নু-লিনাক্স সিস্টেমটা ইনস্টল হয়েছিল, তখনই প্রাথমিক 'fstab' ফাইলটাকে বানিয়ে '/etc' ডিরেক্টরিতে রেখেছিল সিস্টেম। প্রতিবার তার প্রয়োজন অনুযায়ী সেটাকে পড়ে। আপনি যদি কোনো বদল ঘটান, আপনার ইচ্ছা বা প্রয়োজন মোতাবেক, সেই বদল অনুযায়ী তাকে পড়বে। আপনি যদি কখনো ছোটখাটো পার্টিশন কাঠামো বদলান আপনার হার্ডডিস্কের, সেটাকেও আপনাকে নিজেই লিখে দিতে হবে। বা, যদি চান, কোনো একটা পার্টিশন মাউন্ট হবে কি হবে না, সেটাকেও আপনার বলে দিতে হবে। যেমন দেখুন, চার থেকে সাত নম্বর লাইনে, 'options' অংশে, 'rw,noauto,user,exec' এই অংশটা, বা এর নানা রকমফের, আমার যোগ করে দেওয়া। 'rw' মানে ফাইল পড়া আর লেখা দুটোই যাতে করা যায়। 'noauto' মানে আপনা থেকে সিস্টেম মাউন্ট করবে না, আলাদা করে কমান্ড দিয়ে করে নিতে হবে। 'user' মানে শুধু রুট নয়, অন্য কোনো সাধারণ ব্যবহারকারীও মাউন্ট করতে পারবে। '/dev/hdb5' পার্টিশনের বেলায় দেখুন এই 'user' অংশটা নেই, যাতে কখনো ভুলবশত এই পার্টিশনের কোনো ফাইল কেউ বদলে না-ফেলে। 'exec' অংশের মানে কোনো বাইনারি ফাইল চালানো যাবে, এবং 'noexec' মানে তার ঠিক উল্টো। মাউন্টের প্রতিটি অপশানেই তাই, একটা অপশানের ঠিক উল্টোটা হল সেই অপশানের আগে 'no' যোগ করা।

এবার দেখুন তো আমার মেশিনের '/mnt/slackware/etc' ডিরেক্টরি থেকে অন্য 'fstab' ফাইলটা এবার পড়ে দেখুন তো, কিছু বুঝতে পারেন কিনা, তফাতগুলো কোথায় ঘটছে, এবং কেন।

```
#device       mountpoint        fs          options              dump   fsck
/dev/hdb2     swap              swap        defaults             0      0
/dev/hda6     /                 reiserfs    defaults             1      1
/dev/hda1     /mnt/windows/c    vfat        noauto,user,rw       1      0
/dev/hda5     /mnt/windows/d    vfat        noauto,user,rw       1      0
/dev/hdb5     /mnt/arkive       reiserfs    noauto,user,ro       0      0
/dev/cdrom    /mnt/cdrom        iso9660     ro,noauto            0      0
/dev/cdwri    /mnt/cdwri        iso9660     ro,noauto            0      0
/dev/fd0      /mnt/floppy       auto        noauto,owner         0      0
devpts        /dev/pts          devpts      gid=5,mode=620       0      0
proc          /proc             proc        defaults             0      0
```

তাকান, এখানেও একটা '/' পার্টিশন আছে, কিন্তু তার ডিভাইসটা আগের ফাইল থেকে বদলে গেছে, এখন রুট পার্টিশন হয়েছে '/dev/hda6'। কারণ, স্ল্যাকওয়ার সিস্টেম যখন বুট করে, তার রুট পার্টিশন হয় ওটাই। তখন সেই রুট পার্টিশনে যেটা '/etc' ডিরেক্টরি, সেই ডিরেক্টরির '/etc/fstab' ফাইলটাকে তার স্বাভাবিক জায়গাতেই পাই। কিন্তু সুজে দিয়ে বুট করার পর পাই '/mnt/slackware/etc' ডিরেক্টরির ভিতর '/mnt/slackware/etc/fstab' ফাইল হিশেবে। দুটো 'fstab' ফাইল মিলিয়ে দেখুন, সোয়াপ পার্টিশন একই আছে। সুজে আর স্ল্যাকওয়ার দুটো অপারেটিং সিস্টেমই এই সোয়াপ পার্টিশনটা ব্যবহার করে, যখন সুজে দিয়ে বুট করি তখন সুজে, যখন স্ল্যাকওয়ার দিয়ে বুট করি তখন স্ল্যাকওয়ার। এবং সুজের যেটা রুট পার্টিশন, '/dev/hdb3', সেটা এই 'fstab' ফাইলে উল্লিখিতই নেই, তার কারণটা তো আগেই বলেছি, স্ল্যাকওয়ারে যে কারনেলটা আমি এখন ব্যবহার করি তার মধ্যে এক্সএফএস ড্রাইভারটা নেই, তাই সে মাউন্ট করতে পারেনা এই পার্টিশনে। ঠিক সেই একই কারণে '/dev/hdb1' পার্টিশনটাও নেই। দুটো লাইন তাই কমে গেছে স্ল্যাকওয়ারের এই 'fstab' ফাইলে। অন্য তফাতগুলো নিজেই বুঝে নিতে পারবেন, শুধু একটা জিনিষ, স্ল্যাকওয়ার অনেক বিশুদ্ধতাপন্থী, তাই ডিফল্ট রকমে সে সিডিরমে শুধু 'iso9660' ফাইলব্যবস্থার মত করেই পড়ে, দরকার পড়লে রকরিজ বা জলিয়েট রকমে পড়ানো যায়। কী ভাবে? সেটা আমার চেয়ে এখন আপনারই ভালো জানার কথা। আমি সেই কবে পড়েছি, আপনি এই এইমাত্র পড়লেন। কিন্তু পড়বেন কোথা থেকে? এখানে তো দেখাই যাচ্ছে, একটা ডিস্ট্রো থেকে আর একটা ডিস্ট্রোর ডিফল্ট মানে স্বাভাবিক রকমটা আলাদা। এই টুকিটাকিগুলো আপনাকে পড়তে হবে সেই ডিস্ট্রিবিউশনের নিজের ডকুমেন্টেশন থেকে। খেয়াল রাখবেন, একটা অপারেটিং সিস্টেম থেকে আর একটা অপারেটিং সিস্টেমে এই ছোটখাট তফাতগুলো থাকেই।

৫।। একটু ছোট্ট কুপথ — ফাইল বদলানো

একটু কুপথে ঘুরে আসি, ডাইগ্রেস করে আসি, চলুন, তারপর ফের আবার আমাদের ফাইলসিস্টেমের আলোচনার লাস্ট ল্যাপে আসা যাবে। আগের সেকশনে বারবার বলছিলাম ফাইল বদলানো, সেখানে একটা জিনিষ বদলে অন্য জিনিষ লিখে দিতে হবে, ইত্যাদি। কিন্তু সেটা করবেন কী করে? আমাদের আলোচনার শূন্য নম্বর দিনে আমরা প্রচুর কথা বলেছিলাম বিশুদ্ধ টেক্সট এডিটর আর ওয়ার্ড প্রসেসর নিয়ে। তারপর আবার সেই আলোচনা এসেছিল এক আর দুই নম্বর দিনে কমান্ড এডিটরের প্রসঙ্গে। এইবার আপনার সেই কমান্ড এডিটরদের একটাকে লাগে। বিল জয়ের এক্স (ex) তথা ভিআই (vi) এডিটর আর আরএমএসের ইম্যাক্সের (emacs) কথা মনে আছে? ভিআই এখন আরো অনেক উন্নত। তার নাম ভিম (vim), ভিআই-ইমপ্রুভড, উন্নতি এবং বলশালীতা বোঝাতে দীর্ঘ-ঈকার লাগানোই যায়। তাতে দ্বিতীয় পাণ্ডবের মত আগ্রাসী হয়ে ওঠেনা, র‍্যাম সিপিইউ ইত্যাদি অত্যন্ত কম খায় এই ভীম। এ ব্যাপারে আপনি লাগ-এর অরিজিতকে মেল করতে পারেন। ভীমগীতা গেয়ে ও বেশ নাম কিনেছে। আমি নিজে এই কাজে ইম্যাক্স ব্যবহার করি। ধরুন 'fstab' ফাইলটা বদলাতে হবে। '`emacs /etc/fstab`' কমান্ড দিয়ে ফাইলটাকে খুলি, যা দরকার এডিট করি। তারপর '<Ctrl><S>' মেরে সেভ করি, এর মানে কন্ট্রোল সুইচটা টিপে রেখে 'S' সুইচটা টেপা, এবং কাজ শেষ হলে '<Ctrl><X>' মেরে বেরিয়ে আসি। এটা কিন্তু রুট হয়ে করতে হয়। সি শিখতে গিয়ে যে মকশো করার প্রোগ্রাম বানাতে হয়, সেগুলো লিখি এবং কম্পাইল করি ইম্যাক্সে। আমার কাছে এর খুব কোনো উদ্দেশ্য নেই, ডিপ্রেশন কাটানো ছাড়া। রান্না করা আর প্রোগ্রাম লেখা এ দুটোই দেখেছি খুব কাজ দেয়। তবে রান্না করতে গেলেই বাজার করা লাগে, যেটা বোর কাজ এবং খরচেরও। আর প্রোগ্রাম লিখতে কম্পাইল করতে চালাতে আলাদা কোনো খরচই নেই, টেবিলে বসেই করা যায়। ইম্যাক্সের কাছে এই কাজটুকু হল একটা তিমি মাছের পাখনা নাড়ানোর মত। কিন্তু ইম্যাক্স ঠিক ভাবে শেখাটা নিজেই বিরাট একটা কাজ। ইম্যাক্সে ঢুকে আপনি যদি '<Ctrl><H>' মারেন তাহলে আপনি ইম্যাক্সের হেল্পে ঢুকতে পারবেন, ঢুকে পড়তে শুরু করুন, যখন যতটুকু দরকার। একসঙ্গে গোটাটা পড়া একমাত্র কোনো নিরক্ষরের পক্ষেই সম্ভব, শুধু পেজ ডাউন করে গেলেই হয়। এই ভিম আর ইম্যাক্স ছাড়াও জো (joe) আছে, সাতকেলে এড (ed) আছে। এদের শেখা শুরু করার জন্যে ম্যানপেজ আছে।

এবার ধরুন, আপনি আপনার সিস্টেমে দেখলেন কোনো কোনোটা নেই, এদিকে আপনি সেটা দিয়ে কাজ করতে চান। তখন আপনার ডিস্ট্রিবিউশন সিডি, যা দিয়ে আপনি ইনস্টল করেছিলেন, সেটায় খুঁজে দেখুন। এরপরেও হয়ত দেখলেন নেই, বা যে কাজটা করার জন্যে যে প্যাকেজের যে ভার্শনটা চাইছেন সেটা নেই। ধরুন যেমন 'rman', একটু আগে যেটা ব্যবহার করার কথা বললাম, সেটা সব ডিস্ট্রোয় থাকেনা। বা সমগ্র হাউটু-টা (howto)। বা অর্বুদ অর্বুদ লিনাক্সীর প্রাণের চেয়ে প্রিয় এমপ্লেয়ার। সেগুলো কী করে পাবেন? নেটে যদি পেতে চান, তার একটা ভালো জায়গা 'www.freshmeat.net'। বা 'www.sourceforge.net' । এই সাইটে গিয়ে আপনার সাধের প্যাকেজটার নাম দিয়ে সার্চ দিন। তারপর হাইপার লিংক দিয়ে সেটাকে ডাউনলোড করুন। এই ইনস্টলার ফাইলগুলো সচরাচর '*.rpm' বা '*.tar.bz2' বা '*.tar.gz' বা '*.tgz' হয়।

এই আরপিএম ব্যাপারটা চালু করেছিল রেডহ্যাট, আরপিএম এসেছে রেডহ্যাট প্যাকেজ ম্যানেজমেন্ট থেকে। পরে এই ধরনের ফরম্যাট ম্যানড্রেক বা সুজে ব্যবহার করে। আরো কেউ কেউ করে। এই '*.rpm' ফাইলগুলো ইনস্টল করার জন্যে মূল একটা কমান্ড হল 'rpm'। 'rpm' কমান্ডের ম্যানুয়াল পড়ে নিন। যেমন আপনি রুট হয়ে কমান্ড দিতে পারেন '`rpm -Uvh mundu.rpm`'। এটা আপনার সিস্টেমে 'mundu.rpm' প্যাকেজটাকে ইনস্টল করে দেবে, মানে 'mundu' নামের প্রোগ্রামটাকে। বা, আগেই ইনস্টল করা থাকলে আপডেট করে দেবে। যারা একটা কার্যোপযোগী মুণ্ডের অভাবে ভোগেন, এই প্যাকেজটা তাদের খুবই কাজ দেয় বলে শুনেছি। '*.tgz' হল স্ল্যাকওয়ারের নিজের প্যাকেজ বানানোর স্টাইল। অন্য অনেক ডিস্ট্রোতেও ব্যবহার করা যায়। একধরনের প্যাকেজ থেকে অন্য ধরনের প্যাকেজে বদলে নেওয়ার একটা ভালো প্রোগ্রাম আছে, তার নাম 'alien'। '*.deb' হল ডেবিয়ান ডিস্ট্রোর প্যাকেজ বানানোর রকম। যে ডিস্ট্রো আপনি ব্যবহার করছেন তার থেকে ভালো করে পড়ে নিন।

এই যে ধরুন একটা প্রোগ্রাম আপনি ইনস্টল করছেন, এর মানে কী? এই প্রোগ্রামটা লেখা হয়েছে কিছু কোড দিয়ে। সেই কোডটা কম্পাইলড হয়ে বাইনারি ফাইল তৈরি হয়। এই বাইনারি ফাইলগুলো হল এক্সিকিউটেবল, এদের চালানো যায়। এবার কোডকে কম্পাইল করে বাইনারিটা তৈরি হয়ে গেছে, একে বলে প্রিকম্পাইলড বাইনারি। এই

আরপিএম ব্যাপারটা হল তাই। আর মূল কোডটাকে নিজের মেশিনে চালিয়ে কম্পাইল করে নেওয়াও যায়। কম্পাইল করার জন্যে দরকার যে কম্পাইলারটা, তাদের একটা গোটা সংগ্রহ তো যে কোনো গ্নু-লিনাক্স ডিস্ট্রিবিউশনের এমনিতেই দেওয়া থাকে — জিসিসি (gcc)। এবার মূল কোডটাকে কম্পাইল করে নিলেই হল। এই কোডগুলো দেওয়া থাকে '*.tar.bz2' বা '*.tar.gz' ফাইলে। '*.tar.*' মানে সিন্দুক ফাইল, টার প্যাকেজ দিয়ে বানানো, আগেই বলেছি। আর 'bz2' মানে তাকে কোঁকড়ানো হয়েছে 'bzip' প্রোগ্রাম দিয়ে, আর 'gz' মানে কোঁকড়ানো হয়েছে 'gzip' দিয়ে। ম্যানপেজ পড়ে নিন। নানা ভাবে এদের নিয়ে কাজ করা যায়, ইউনিক্স তথা গ্নু-লিনাক্সের দর্শনই তাই, কোনো একটা কাজ সবসময়েই একাধিক ভাবে করা যায়। 'tar' দিয়ে সরাসরি এই 'bz2' বা 'gz' ফরম্যাটে বানানো কোঁকড়ানো সিন্দুক খোলার উপায় আছে। ধরুন ওই 'mundu.rpm' এবার আপনি আর 'rpm' মানে প্রিকম্পাইলড বাইনারিতে না-নামিয়ে কোঁকড়ানো সিন্দুকে নামিয়েছেন। আগেরবার ইনস্টল করার পর থেকে বহুত গন্ডগোল করছিল, আপনার নৃমুণ্ডের আবছায়া তৈরি হচ্ছিল, আলো অন্ধকারে যেখানেই যান, এবার আপনি আর মুন্ডু নিয়ে ওইসবে রিস্কে যেতে চাননা। আপনার মুন্ডু যাতে আপনার হার্ডওয়ারের সঙ্গে খাপে খাপে মিলে যায়, আপনার হার্ডওয়ারের সফটওয়ারের প্রতিটি খুঁটিনাটি নিয়ে আপনার সিস্টেমেই কম্পাইলড হয়, সেই ব্যবস্থাই এবার করতে চান, মানে সরাসরি ধরো আর কম্পাইল করো। ও নিজেই আপনার সিস্টেমের খুঁটিনাটি অনুযায়ী নিজেকে বদলে নেবে। ফাইলটা যদি 'mundu.tar.bz2' হয় তাহলে কমান্ড দিতে হবে, '`tar xvjf mundu.tar.bz2`'। এতে 'mundu.tar.bz2' ফাইলটা কোঁকড়ানো অবস্থা থেকে বেরিয়ে একটা স্বাভাবিক টার-সিন্দুকে পরিণত হল, এবং তারপর একটা 'mundu' নামের ডিরেক্টরি বানিয়ে, ওই সিন্দুক খুলে সমস্ত ফাইলকে সেই 'mundu' ডিরেক্টরিতে রাখা হয়ে গেল ওই একটা কমান্ড দিয়েই। যদি ফাইলটা 'mundu.tar.gz' হত তাহলে আপনার কমান্ডটা সামান্য একটু বদলে যেত, হত '`tar xvzf mundu.tar.gz`'। এতেও ওই একই ভাবে আপনি 'mundu' বলে একটা ডিরেক্টরি পেতেন, যার মধ্যে ফাইলে বা সাবডিরেক্টরিতে ভরা থাকত আপনার প্রয়োজনীয় কোডগুলো।

এবার ওই 'mundu' ডিরেক্টরিতে ঢুকে আপনি পাবেন একটা 'README' ফাইল বা একটা 'INSTALL' ফাইল বা দুটোই। এদের 'less' দিয়ে পড়ে নিলেই আপনি জানতে পারবেন কী করতে হবে। করাটাও এমন হাতি ঘোড়া কিছু না। এরা প্রত্যেকেই ব্যবহার করে জিসিসির মত আর একটা গ্নু পরশপাথর, যার নাম মেক (make), শীতকাল চলে যাচ্ছে দ্রুত গিয়ে খেজুরগাছে হাঁড়ি বেঁধে আসুন। এই 'make' যাতে কাজ করতে পারে তার জন্যে একটা 'makefile' দেখুন রয়েছে ডিরেক্টরিতে। এবার আপনাকে কমান্ড দিতে হবে, '`./configure`'। এই '`.`' মানে কারেন্ট ডিরেক্টরি, যে ডিরেক্টরিতে এই মুহূর্তে আপনি আছেন, সেখানে থাকা '`configure`' নামের প্রোগ্রামটা চালাও। এই কনফিগার প্রোগ্রামটা চালিয়ে ওই সফটওয়্যার প্যাকেজ ইনস্টল হওয়ার আগে আপনার সিস্টেমের প্রত্যেকটা প্রয়োজনীয় খুঁটিনাটি কনফিগারেশন জেনে নেবে। তারপর, কনফিগারেশনের কাজ শেষ হলে, কমান্ড প্রম্পট ফেরত এলে কমান্ড দেবেন '`make`'। অনেকসময় কনফিগার প্রোগ্রামটা থাকেনা, সরাসরি মেকফাইল থাকে। তখন মেক দিয়েই কাজ শুরু করতে হয়। এতে ওই রাশিরাশি কোডের ঠিক ঠিক প্রয়োজনীয় অংশ আপনার সিস্টেমের মানানসই রকমে সাজিয়ে দেবে। এবার মেকের কাজ শেষ হলে রুট হবেন আপনি আর কমান্ড দেবেন '`make install`'। গোটা কাজটা একবার কমান্ড দিয়েও করা যেত, '`./configure && make && make install`'। এই '`&&`' অংশটা ব্যাশকে বলে দিত, আগের কাজটা শেষ হলেই কেবল পরের কাজটায় যেও, ব্যাশের ম্যানপেজ থেকে মিলিয়ে নিন। এরপর বেশ কিছুটা সময় ধরে কনসোলের পর কনসোল জুড়ে রাশিরাশি প্রোগ্রামিং কচকচি ফুটে উঠতে থাকবে। তারপর কমান্ড প্রম্পট ফেরত আসবে। আর কী, খেল খতম, কম্পাইলেশন হয়ে গেল। এখুনি যান, তাদের প্রত্যেককে একটা করে মেল করে দিন, আসুন, যারা এতদিন ধরে ব্যাঙ্গবিদ্রূপ করে আসছে যে আপনার কোনো মুন্ডু নেই। মুন্ডু হাত পা যা যা খুশি এভাবে নেট থেকে নামান আর ইনস্টল করে নিন। শুধু টেলিফোন বিলটা মাথায় রাখলেই হল।

এবার আমরা যাব আমাদের সেই দীর্ঘ প্রতীক্ষার গ্নু-লিনাক্স ফাইলসিস্টেম হায়েরার্কিতে। শুধু একটা কথা বলতে ভুলে গেছি, ৪ নম্বর সেকশনে, সেটা হল, ওই 'mount' কমান্ডটার একটা বিপরীত কমান্ড আছে 'umount', যা মাউন্টের ঠিক উল্টোটা করে, মানে আনমাউন্ট। আদতে কমান্ডটার নাম ছিল 'unmount', ১৯৭০ অব্দি ইউনিক্স জগতে এই চেহারাতেই তাকে পাওয়া যেত, তার পর কবে 'n'-টা খসে গেছে, দীর্ঘ কঠোর তদন্তের পরেও জানা যায়নি 'n'-এর এন্ড ঘটল কিভাবে, কোন আততায়ীর হাতে।

৬।। গ্নু-লিনাক্স ফাইলসিস্টেম হায়েরার্কি

আগের দুদিন থেকে আজ এই দীর্ঘ আলোচনার মধ্যে দিয়ে আমাদের কাছে হার্ড ডিস্কের ভৌত এবং মূর্ত ভূমি, সেখানে তথ্য তথা ফাইল এবং ডিরেক্টরি ইত্যাদি নানা জাতের তথ্য-সমাহার রাখার প্রক্রিয়া, আর সেই তথ্যকে একটা কেজো চলন্ত কম্পিউটারের কাজের সম্পর্কে মিলিয়ে নেওয়ার জন্যে ফাইল আর ডিরেক্টরি দিয়ে সাজানো একটা বিমূর্ত ভূমি — এই তিনের আলাদা অস্তিত্ব এবং একই সঙ্গে অঙ্গাঙ্গী সম্পর্ক — এই গোটাটাই এখন আমাদের কাছে পরিচিত। মূর্ত ভূমিটা হল সেই কাঁচা ডিস্ক যা আমরা কিনে আনি। আর রুট ডিরেক্টরি থেকে হোম বিন এটসেট্রা ইত্যাদিতে বিন্যস্ত কেজো কম্পিউটার ভূমিটা হল বিমূর্ত জায়গা যা আমাদের তৈরি করে নিতে হয়। বিমূর্ত এবং সাঙ্কেতিক, বহুবার বলেছি আমরা, অ্যাবস্ট্রাক্ট এবং সিম্বলিক। এই জায়গাটা অত্যন্ত গুরুত্বপূর্ণ, কারণ যে প্রোগ্রামগুলো নিয়ে আমরা কাজ করি তারা কেউ কাজ করে একদম ফাইল নিয়ে, যেমন ধরুন একটা ‘emacs’ বা ‘mkdir’, আবার কেউ কাজ করে একদম কাঁচা ভৌত ভূমি নিয়ে যেমন ‘fdisk’ বা ‘fsck’। আমরা যে প্রোগ্রামগুলোকে চিনি এবং আমাদের কম্পিউটার সময়ের প্রায় গোটাটাই যাদের নিয়ে কাজ চলে তারা প্রায় সবাইই কাজ করে বিমূর্ত ভূমিটায়। কিন্তু এই বিমূর্ত ভূমিটা দাঁড়িয়ে আছে তথ্য আর তথ্য-সমাহারের অনেকগুলো প্রক্রিয়ার উপর, মানে ফাইলসিস্টেম। এই ফাইলসিস্টেমগুলোর উপর দাঁড়িয়ে এরা কাজ করে, তাই এদের সঙ্গে মিলিয়ে না-নিলে গোটা প্রক্রিয়াটাই বোঝা সম্ভব না।

ইতিমধ্যেই কাজের ওই বিমূর্ত ভূমিটার সঙ্গে অনেকটাই পরিচিত, যেটা একটা ঐক্যবদ্ধ হায়েরার্কিন্যস্ত ফাইলসিস্টেম, ডিরেক্টরি সাবডিরেক্টরি ফাইলের ক্রমানুসারী সমগ্র, যা তৈরি হয়েছে বিভিন্ন তথ্য রাখার প্রক্রিয়া মানে ফাইলসিস্টেম নিয়ে। এখন থেকে আগের বাক্যে ফাইলসিস্টেম কথাটার দ্বিতীয় ব্যবহারটাকে আমরা আপাতত ভুলে যাব, যেখান থেকে শুরু করেছিলাম ছয় নম্বর দিনে, সেখানে ফেরত যাব, ফাইলসিস্টেম মানেই ডিরেক্টরি আর ফাইলের ওই ঐক্যবদ্ধ সমগ্রটা। সেটার হায়েরার্কিটাকেও চিনি আমরা। রুট ডিরেক্টরি মানে ‘/’ থেকে শুরু। তার মধ্যে নানা ডিরেক্টরি। তাদের মধ্যে আবার ডিরেক্টরি, আবার, এই ভাবে চলে। প্রতিটি ডিরেক্টরিতেই থাকতে পারে আরো নিচের ডিরেক্টরি, মানে সাবডিরেক্টরি, যতখুশি, এবং যতখুশি ফাইল। এবার, এই ক্রমটা আছে বলেই, ঐক্যটা আছে বলেই, একটা ফাইলসিস্টেমে প্রতিটা ফাইল এবং ডিরেক্টরির একটা ঠিকানা আছে — একটা বিমূর্ত ঠিকানা, রুট ডিরেক্টরি থেকে ঠিকানাটা শুরু হয়ে শেষ হয় যে ডিরেক্টরিতে ফাইলটা আছে সেই ডিরেক্টরির নামে এসে (মূর্ত ঠিকানাটা, খেয়াল করুন, আর আমাদের আপাতত দরকার পড়বেনা, মানে কত নম্বর সিলিন্ডারের কত নম্বর সেক্টরের কত নম্বর ব্লক ইত্যাদি)। যেমন ধরুন ‘/mnt/slackware/etc/fstab’ ফাইলটার ঠিকানা হল ‘/’ ডিরেক্টরির মধ্যে ‘mnt’ ডিরেক্টরির মধ্যে ‘slackware’ ডিরেক্টরির ‘etc’ ডিরেক্টরির মধ্যে ‘fstab’ নামের ফাইল। আমি এবং আমার জ্যান্ত অপারেটিং সিস্টেম দুজনেই এটাকে চিনে নিতে পারব বলামাত্র। এবার পরের আলোচনায় যাওয়ার আগে একবার, মন দিয়ে, ছয় নম্বর দিনের ৪ নম্বর সেকশনের আর সাত নম্বর দিনের ১১ নম্বর সেকশনের ফাইলসিস্টেমের ছবিদুটো মিলিয়ে মিলিয়ে, তাদের পার্থক্য এবং মিল সহ, ভালো করে দেখে নিন। এবার আমরা গ্নু-লিনাক্সে ফাইলসিস্টেম হায়েরার্কির ওই ছকটা বুঝব।

ওই ছকটা কিন্তু একদিনে গড়ে ওঠেনি। ‘৯০-এর দশকের মাঝামাঝি অব্দি, গ্নু-লিনাক্স বয়োপ্রাপ্ত হয়ে ওঠার আগে অব্দি প্রতিটি আলাদা আলাদা ডিস্ট্রো ছকটা বানিয়ে নিত নিজের মত করে। এতে সমস্যাটা বুঝতেই পারছেন। একটা ছকে অভ্যস্ত কেউ অন্য ছকে কাজ করতে গেলে ছক বোঝাটাই তার কাজ হয়ে দাঁড়াবে। তার চেয়েও সমস্যা অবলা কোডগুলোর। তাদের বানানো হয়েছে একটা ছকের সঙ্গে মিলিয়ে — এই ডিরেক্টরিতে খোঁজো এই ফাইলটা আছে কিনা, ওই ডিরেক্টরিতে খোঁজো ওই ফাইলটা আছে কিনা, ইত্যাদি। তারা আর অন্য ছকের কোনো ডিস্ট্রোতে কাজই করবে না। প্রতিটা ডিস্ট্রোর আলাদা আলাদা প্রোগ্রাম বানাতে হবে। ইউনিক্সের জগতে পোজিক্স যে কারণে এসেছিল। সেই পোর্টেবিলিটি, স্থানান্তরযোগ্যতা, আমাদের এই আলোচনায় আমরা অনেকটা সময় ব্যয় করেছি সেটা বুঝতে। এই জন্যে এল একটা ডকুমেন্ট, তার নাম ‘লিনাক্স ফাইলসিস্টেম স্ট্রাকচার’ এবং সেই ডকুমেন্টটা বানানোর জন্যে একটা গ্রুপ — এই ডকুমেন্ট আর গ্রুপ দুটোকেই ডাকা হয় এফএসস্ট্যান্ড (FSSTND) বলে, ফাইলসিস্টেম স্ট্যান্ডার্ড। এই এফএসস্ট্যান্ড-এর কাজটাই বাড়তে বাড়তে পরে ড্যানিয়েল কুইনলানদের ওই ফাইলসিস্টেম হায়েরার্কি স্ট্যান্ডার্ডে পৌঁছয়, যার কথা আমরা আগেই বলেছি। এই কাজটা কিন্তু এখনো সম্পূর্ণ নয়, এখনো অনেক না-মেটা মনান্তর

আছে। যেমন যেসব স্ক্রিপ্ট (মানে শেলে কাজ করা প্রোগ্রাম) কোনো একটা বিশেষ নির্মাণের কম্পিউটারের সঙ্গে সম্পৃক্ত নয়, যে কোনো মেশিনেই চালানো যায় যাদের, তারা কি '/usr/share' ডিরেক্টরিতে থাকবে না থাকবে-না? এই '/usr/share' ডিরেক্টরিটাকে কি আরো সাবডিরেক্টরিতে ভাঙা হবে, ইত্যাদি। এরকম আরো কিছু। একটা ডিস্ট্রো থেকে আর একটা ডিস্ট্রো-তে কিছু কিছু তফাত তো প্রথম দৃষ্টিতেই পরিদৃশ্যমান। যেমন, বহিরাঙ্গিক পার্টিশনগুলোকে, যেমন আমার মেশিনে ওই চারটে পার্টিশন — '/dev/hdb5', '/dev/hda6', '/dev/hda1', '/dev/hda5', মানে আর্কাইভ, স্ল্যাকওয়ার, আর উইনডোজ পার্টিশনদুটো — এদের কোন মাউন্টপয়েন্টে মাউন্ট করা হবে এর কোনো ধরাবাঁধা সংস্থান এখনো নেই। এইরকম আরো আছে। একাধিক ডিস্ট্রোয় কাজ করতে গেলেই দেখবেন। এবং এখন আমরা যে ছকটা দেব সেটা যে প্রত্যেকটা ডিস্ট্রো হুবহু মেনে চলে তা কিন্তু নয়। এবার ছকটা দেওয়া যাক।

৬.১।। রুট ডিরেক্টরি বা '/'

এফএসস্ট্যান্ড মানার জন্যে রুট ডিরেক্টরিতে কিছু ডিরেক্টরি থাকতেই হবে, বা মূল ডিরেক্টরিটা অন্য কোথাও থাকলেও তার সিম্বলিক লিংক এখানে থাকতেই হবে, এরা হল রুট ডিরেক্টরির ন্যূনতম বাসিন্দা। গ্নু-লিনাক্সের ফাইলসিস্টেম স্ট্যান্ডার্ড অনুযায়ী এদের থাকতেই হবে। এর বাইরে আদতে আরো বহু কিছু থাকে। তাদের কথায় আসছি আমরা।

/bin — এই বিন ডিরেক্টরিতে থাকবে অত্যাবশ্যক কমান্ড বাইনারিগুলো, যাদের কমান্ড প্রম্পটে টাইপ করে তারপর এন্টার মেরে আমরা চালাই। আমরা কমান্ড প্রম্পটে কাজ করার জন্যে যে আদেশগুলো নিয়ে এই পুরো পাঠমালায় আলোচনা করেছি, তাদের প্রায় সকলেরই আবাস এই বিন ডিরেক্টরি। বিন নিয়ে আলোচনায় আসছি।

/boot — এই বুট ডিরেক্টরিতে থাকে সিস্টেম বুট করার জন্যে প্রয়োজনীয় কারনেল ইমেজ এবং তার কিছু আনুষঙ্গিক ফাইল। তারা যদি হুবহু এখানে না-থাকে, থাকে তাদের সিম্বলিক লিংক। কারনেলটা ব্যবহার করতে গেলে যে মডিউলগুলো লাগবে তাদের তালিকাও দেওয়া থাকে এখানেই। যেমন '/boot/vmlinuz'। শুধু বুট-প্রক্রিয়ার কনফিগারেশন ফাইলগুলো এখানে থাকেনা, দেখেছেন আগেই, লিলোর কনফিগারেশন ফাইল 'lilo.conf' থাকে '/etc' ডিরেক্টরিতে। তাকে যখন '`lilo`' কমান্ড দিয়ে ইনস্টল করা হয় সে কিছু ফাইল তৈরি করে এই বুট ডিরেক্টরিতে। যেমন '/boot/boot.0300' মানে মাস্টার বুট রেকর্ডের ব্যাকআপ। বা '/boot/boot.b' মানে মূল বুট সেক্টর বা তার সিম্বলিক লিংক।

/dev — এই ডেভ ডিরেক্টরিতে থাকে রাশিরাশি ডিভাইস ফাইল, যাদের নিয়ে যথেষ্ট বিশদভাবে আলোচনা হয়েছে আমাদের ছয় নম্বর দিনে।

/etc — এই ইটিসি বা এটসেট্রা ডিরেক্টরিতে থাকে একটা সিস্টেমের হার্ডওয়ার এবং সফটওয়ার সংক্রান্ত প্রায় সমস্ত কনফিগারেশন ফাইল। এর কিছু ফাইলকে এর মধ্যেই বেশ ধরে ধরে পড়েছি আমরা, বারবার। সিস্টেমের স্নায়ুকেন্দ্র বলা চলে এই '/etc' ডিরেক্টরিকে। আমরা এর কথায় পরে আসছি।

/lib — এই লিব ডিরেক্টরিতে থাকে লাইব্রেরি ফাইলগুলো, যাদের কথা আমরা আগেই বলেছি, প্রোগ্রামগুলো এই লাইব্রেরিদের কাজে লাগায়। প্রতিটি প্রোগ্রামের জন্যে আলাদা আলাদা ফাংশন বা ক্রিয়া না লিখে, সব প্রোগ্রামের জন্যে একত্রে এই লাইব্রেরি ফাংশনগুলোকে তুলে রাখা হয় লিব ডিরেক্টরিতে। আর থাকে কারনেল মডিউলগুলো, মানে, কারনেলের চলার জন্যে ছোট ছোট টুকরো প্রোগ্রাম বানানো থাকে, এক ধরনের ড্রাইভার, এক এক ধরনের উপাদানের জন্যে। ধরুন একটা বিশেষ সাউন্ডকার্ড আপনি ব্যবহার করছেন, তার জন্যে কারনেলে একটা বিশেষ মডিউল লাগিয়ে নিতে হচ্ছে। আবার সাউন্ডকার্ডটা বদলালেন, মডিউলটা বদলে নিতে হল।

/mnt — এই এমএনটি বা মাউন্ট ডিরেক্টরির সঙ্গে আপনারা ইতিমধ্যেই বারংবার পরিচিত হয়েছেন। বহিরাগত পার্টিশনদের বসতে দেওয়ার জেনেরাল শীতলপাটির মত। তবে সব ডিস্ট্রো এটা হুবহু মানেনা। ম্যানড্রেক স্ল্যাকওয়ার খুব মানে। কিন্তু সুজে নানা অন্য অন্য জায়গা বানিয়ে দেয়। যেমন আমার উইনডোজ ডিরেক্টরি দুটোর জন্যে নিজে থেকেই বানিয়ে দিয়েছিল '/data1' আর '/data2' বলে দুটো ডিরেক্টরি, একদম রুট ডিরেক্টরিতে। পরে নিজে বদলে নিতে হয়। তবে খেয়াল রাখবেন, এইসমস্ত যাবতীয় জিনিষ বদলানোর পরেই '/etc/fstab' ফাইল থেকে একবার মিলিয়ে নেবেন, সবগুলো ঠিকঠাক আছে কিনা। নিজের ইচ্ছেমতন বাড়তি মাউন্টপয়েন্ট বানিয়ে নেওয়া যায় এই '/mnt' ডিরেক্টরিতে, আমরা দেখেছি, '/mnt/arkive' বা '/mnt/slackware' ইত্যাদি। আর চালু

কিছু মাউন্টপয়েন্ট এখানে এমনিতেই থাকে, যদি ডিস্ট্রো সেখানে নিজস্ব কেরদানি না-মারে। যেমন '/mnt/cdrom' বা '/mnt/floppy' ইত্যাদি। সুজেতে সেসব থাকে '/media' ডিরেক্টরির মধ্যে।

/opt — অপশনাল বা ঐচ্ছিক, ডিস্ট্রোর ডিফল্ট ইনস্টলেশনের নিজস্ব কাঠামোর বাইরে থেকে যুক্ত প্যাকেজদের জায়গা। যেমন ওপেনঅফিস, মোজিলা ইত্যাদি প্যাকেজগুলো এখানে থাকে। এফএসস্ট্যান্ড অনুযায়ী, সমস্ত বহিরাগত প্যাকেজদেরই এখানে থাকার কথা। আর এই প্যাকেজটা ইনস্টল করার জন্যে আপনার যে আনুষঙ্গিক ফাইলগুলো থাকবে তাদের থাকার জায়গা এই '/opt' ডিরেক্টরির মধ্যেই ওই প্যাকেজের নামে আলাদা একটা ডিরেক্টরিতে। ধরুন ওই 'mundu' প্যাকেজটা, বাইরে থেকে যোগ করা প্যাকেজ, কারণ, ডিফল্টে মানুষের একটা মুন্ডু এমনিতেই থাকার কথা, আপনি যদি ইনস্টল করেন, সেই ইনস্টলেশনটা ঘটবে '/opt/mundu' ডিরেক্টরির মধ্যে। ইনস্টল করতে গিয়ে যা যা বাড়তি ফন্ট মানে অক্ষর-আকার লাগবে, বা ছবি, বা ডেটাবেস, সে সবই থাকবে এই ডিরেক্টরির ভিতর। এই প্যাকেজ যে বাইনারি ফাইল তৈরি করবে সেটা থাকার কথা '/opt/bin' ডিরেক্টরিতে, ডকুমেন্টেশন থাকার কথা '/opt/doc' ডিরেক্টরিতে, ইত্যাদি। তবে, বেশিরভাগ প্যাকেজই এতসব খাবলা দায়িত্ব পালন করেনা।

/sbin — এর মধ্যে থাকে অত্যাবশ্যক সিস্টেম বাইনারিগুলো। গ্নু-লিনাক্সে বাইনারি বা এক্সিকিউটেবল মানে চালানো যায় এমন ফাইলগুলোকে দুটো ভাগে ভাগ করা হয়। একটা সাধারণ বা নরমাল এক্সিকিউটেবল। যা সর্বসাধারণের ব্যবহারযোগ্য। আর অন্যগুলো হল ভিআইপি এক্সিকিউটেবল। সিস্টেম অ্যাডমিনিস্ট্রেশন বা পরিচালনার কাজে ব্যবহারযোগ্য বিশেষ বাইনারিগুলো থাকে এই '/sbin' ডিরেক্টরিতে। এর চেয়ে একটু কম গুরুত্বপূর্ণ সিস্টেম বাইনারিগুলো থাকে '/usr/sbin' ডিরেক্টরিতে বা '/usr/local/sbin' ডিরেক্টরিতে। এই '/sbin' ডিরেক্টরি তথা সিস্টেম বাইনারিদের কথায় আসছি আমরা।

/tmp — অস্থায়ী বা টেম্পোরারি ফাইলদের জায়গা। একটা নির্দিষ্ট সময়ের পরে যাদের আর দরকার পড়বেনা। অনেক প্রোগ্রাম কাজ করার সময় বিশেষ কিছু ফাইল বা ডিরেক্টরি বা প্রোগ্রামকে লক করে দেয়, চলার সময়টুকু জুড়ে অন্য কেউ তাদের স্পর্শ করতে পারবেনা। কিম্বা চলার কাজে কিছু অস্থায়ী তথ্য তৈরি করে, প্রোগ্রাম চলা শেষ হয়ে যাওয়ার পরে যাদের আর কাজে লাগবে না। এই সবকিছুকে অস্থায়ী ভাবে রাখা হয় এই '/tmp' ডিরেক্টরিতে। '`su`' করে রুট হয়ে নিয়ে এখান থেকে ফাইল ওড়ানোই যায়, কিন্তু কদাচ নয়, যদি-না, হুবহু জানা থাকে, ঠিক কী করছি। সেই মুহূর্তে যে প্রসেসগুলো চলছে সিস্টেমে তাদের অনেকেরই অত্যন্ত জরুরি মালপত্তর এখানে থাকে। গোটা সিস্টেম হ্যাং করে যেতে পারে ক্র্যাশ করে যেতে পারে — গ্নু-লিনাক্স সিস্টেমে যা সত্যিই একটা দুর্ঘটনা, এত কম ঘটে। যেমন ওপেনঅফিস ১.০ ভার্শনে আমার এনভিডিয়া ড্রাইভারের সঙ্গে কিছু একটা গোলযোগ হচ্ছিল, পরে, এনভিডিয়া ড্রাইভারের একটা পুরোনো ভার্শন ব্যবহার করে যেটা চলে গেল, সেই সময়ের গোটা পাঁচেক হ্যাং আজ পর্যন্ত এই দু-বছরে আমার পাঁচমাত্র হ্যাংদোলন। নিজে লিখতে গিয়ে তার আগের উইনডোজের বছরগুলোর সঙ্গে মিলিয়ে নিজেরই কেমন অবিশ্বাস হচ্ছে। এই '/tmp' ডিরেক্টরির মোট ফাইলের আয়তন সচরাচর কয়েক কিলোবাইটের বেশি হয়না। বুট বা শাটডাউনের সময় সচরাচর এই ডিরেক্টরিকে ফাঁকা করে দেওয়া হয়।

/usr — মূল '/' হায়েরার্কির মধ্যে দ্বিতীয় একটা হায়েরার্কি বা ক্রম। এই '/usr' ডিরেক্টরিতে থাকে গোটা সিস্টেমের মোট ফাইল এবং ডিরেক্টরির সমগ্রের সিংহভাগ। আমরা এখানে সুজে এবং স্ল্যাকওয়ার দুটো ডিস্ট্রোরই রুট ডিরেক্টরিতে থাকা সাবডিরেক্টরির গুলোর আলাদা আলাদা ভাবে মোট আয়তন আর গোটা সিস্টেমটার মোট আয়তন দিয়েছি, পাশাপাশি। সাধারণ একটা গড় ইনস্টলেশন। তার থেকে সামান্য বেশি হতে পারে, সবসময়ে ডিভলপমেন্টের, মানে কম্পিউটার ভাষার আর প্রোগ্রামিং-এর যে প্যাকেজগুলো এই দুটো সিস্টেমেই ইনস্টল করা আছে, তার সবগুলো সব সিস্টেমে থাকেনা। আর পাবলিশিং সংক্রান্ত প্যাকেজও দু-চারটে বেশি থাকতে পারে। এর বাইরে একটা স্বাভাবিক একটা হোম পিসির একটা সাধারণ ইনস্টলেশনের আয়তনটাও এখান থেকে পেতে পারবেন। যে কমান্ড দিয়ে আমি এ তালিকাটা পেয়েছি সেটা হল '/' ডিরেক্টরিতে দাঁড়িয়ে '`du -chs *`', তারপর তাকে আমাদের চেনা কায়দায় রিডাইরেক্ট করে। ম্যানপেজ পড়ে দেখুন এই 'du' কমান্ডটা ভারি চমৎকার ভাবে ডিস্কভূমির ব্যবহারটা দেখায়। এর '-chs' অপশানগুলোর মানে নিজেই দেখে নিন। এছাড়া আর একটা

কমান্ডও এই কাজে খুব ব্যবহার হয়, 'df'। তালিকাটায় 'M' মানে মেগাবাইট, 'K' মানে কিলোবাইট, 'G' মানে গিগাবাইট। সুজেতে দেখুন '/usr' জুড়ে আছে গোটা সিস্টেমের মোট ব্যবহৃত ডিস্কভূমির প্রায় একাত্তর শতাংশ আর স্ল্যাকওয়ারে প্রায় তিরাশি শতাংশ। এই '/usr' ডিরেক্টরির আলোচনায় আমরা পরে আসছি।

/var — ভ্যারিয়েবল বা পরিবর্তনশীল তথ্যদের আবাস। সিস্টেম লগিং সংক্রান্ত তথ্যের ফাইল, মেল আর প্রিন্টার স্পুলের ফাইল, এই সব থাকে '/var' ডিরেক্টরিতে। এই '/var' ডিরেক্টরি দুটো সিস্টেম কখনো শেয়ার করতে পারেনা, মানে দুজনেই আলাদা ভাবে একই পার্টিশনকে ব্যবহার করতে পারেনা। কারণ এখানে থাকা তথ্যের গোটাটাই হল সিস্টেমের একান্ত নিজস্ব। এই ডিরেক্টরির তথ্যগুলো সিস্টেম ব্যবহারের সঙ্গে তাল মিলিয়ে সবসময়েই বদলাচ্ছে।

সিস্টেম	সুজে	স্ল্যাকওয়ার
bin	7.2M	6.2M
boot	6.7M	2.9M
dev	428K	293K
etc	48M	23M
home	114M	3.1M
lib	64M	19M
mnt	2.5K	3.0K
opt	658M	350M
proc	2.0K	16K
root	2.2M	489K
sbin	11M	6.9M
tmp	8.4M	13M
usr	2.5G	1.9G
var	128M	23M
মোট	3.5G	2.3G

লিনাক্স ফাইলসিস্টেম স্ট্যান্ডার্ড অনুযায়ী উপরের ডিরেক্টরিগুলো একটা রুট ডিরেক্টরিতে থাকতে হবে, বা, নিদেন, তাদের লিংক। কিন্তু এগুলো ছাড়াও আরো কিছু ডিরেক্টরি থাকতে পারে। কোনো কোনো ডিস্ট্রোতে এর বাইরেও ডিরেক্টরি থাকে বৈকি, যেমন সুজেতেই '/srv', যেটার মধ্যে সার্ভার সংক্রান্ত তথ্য রাখে। এর মানে, যতটুকুই হোক, গ্নু-লিনাক্স স্ট্যান্ডার্ড থেকে নড়ছে সুজে। এফএসস্ট্যান্ড অনুযায়ী নিচের ডিরেক্টরিগুলো রুট ডিরেক্টরিতে থাকতে পারে।

/initrd — এই '/initrd' ডিরেক্টরি একটা গ্নু-লিনাক্স সিস্টেমকে দেয় বুটলোডার মারফত র‍্যামডিস্ক ব্যবহারের সুযোগ। র‍্যামডিস্ক (RAM Disk) মানে র‍্যান্ডম-অ্যাকসেস-মেমরির একটা বিশেষ এলাকা, র‍্যামডিস্ক বানানোর সফটওয়ার দিয়ে যাকে বানিয়ে নিতে হয়, যে এলাকাটা ঠিক একটা ছোট্ট সাইজের হার্ডডিস্ক পার্টিশনের মত ব্যবহার করবে। হার্ডডিস্ককে নকল বা এমুলেট করবে এই র‍্যামাঞ্চলটা। সুবিধেটা এই যে হার্ডডিস্ক পার্টিশন থেকে তথ্য পড়ার চেয়ে বহুগুণ বেশি গতিতে এইখান থেকে তথ্য পড়ে নিতে পারে সিপিইউ, দ্রুত তার কাজে ঢুকে যেতে পারে। আর যেই মেশিন শাটডাউন বা রিবুট করছি তখনই এই র‍্যামডিস্কটা মুছে যাচ্ছে। একটা সঠিক সাইজের অপারেটিং সিস্টেমকে, র‍্যাম-এলাকাটার মধ্যে যেটা ভরে যাবে, এইভাবে র‍্যামে রেখেই কাজ করে যেতে পারে কম্পিউটার। '/initrd' ডিরেক্টরিটা সুযোগ দেয় র‍্যামডিস্ককে রুট ফাইলসিস্টেম হিশেবে মাউন্ট করার, এরপর এখান থেকেই প্রয়োজনীয় প্রোগ্রাম চালানো যায়। তারপর, প্রাথমিক কাজ শেষ হলে, সঠিক হার্ডডিস্ক পার্টিশনকে রুট ডিরেক্টরি হিশেবে মাউন্ট করে নেওয়া যায়। এই ভাবে র‍্যামডিস্ক রাখার সুযোগ করে দিয়ে '/initrd' ডিরেক্টরি সিস্টেম বুটের গোটা কাজটাকে দুটো অংশে ভেঙে দেয়, প্রথম অংশটায় কারনেলে অনেক কম ড্রাইভার কম্পাইল করা থাকলেও অসুবিধে নেই, দ্বিতীয় স্তরে প্রয়োজনীয় মডিউলগুলো '/initrd' ডিরেক্টরি থেকে তুলে নেয় কারনেল। এই ডিরেক্টরির জটিলতায় আর গিয়ে কাজ নেই, আপনার ইচ্ছে থাকলে আপনি নিজেই শিখে নিতে পারবেন। '/initrd' বানানোর যে কমান্ড, 'mkinitrd', তার ম্যানপেজ পড়ুন। আমাদের এখানে তুলে দেওয়া তালিকাটা দেখুন, সুজে রুট ডিরেক্টরির মধ্যে কোনো '/initrd' ডিরেক্টরি নেই। এই '/initrd' ডিরেক্টরিটা ছাড়াই সুজে কিন্তু র‍্যামডিস্ক ব্যবহার করছে। দেখুন তো, পাঁচ নম্বর দিনে তুলে দেওয়া '/etc/lilo.conf' ফাইলটা থেকে তার কোনো হদিশ পান কিনা।

/home — বারবার ব্যবহার করতে করতে এই ডিরেক্টরিটা আমাদের চেনা হয়ে গেছে। সমস্ত ব্যবহারকারীর নিজের নিজের হোম ডিরেক্টরি বা নিজের ঘর মিলিয়ে তৈরি এই ব্যবহারকারীদের কোয়ার্টার। এই সার্বজনীন ঘরগেরস্থালির মধ্যে একজন বিশেষ ব্যবহারকারী বা ইউজারের হোম ডিরেক্টরির হল '/home/$USER' যার অন্য নাম '~/'। এই দুটোকে কি চিনতে পারলেন? যদি না পারেন, তার মানে আপনি পুরোনো পড়াগুলো ভালো করে তৈরি রাখছেন না। যান, সাত নম্বর দিনের ৯.৩ নম্বর সেকশনটা আর একবার ভালো করে পড়ে আসুন। প্রত্যেকটা ইউজারের নিজের ঘরে কী কী ফাইল থাকে তার সঙ্গে মোটামুটি একটা পরিচয় তো আমাদের আগেই হয়ে গেছে। কিছু ডিরেক্টরি, কিছু ফাইল, আর কিছু সেই সতত-অদৃশ্য ডটানন ফাইল এবং ডিরেক্টরি, যাদের মধ্যে রাখা আছে

বিভিন্ন ধরনের কনফিগারেশন বা সিস্টেম সেটিং। মনে করতে পারছেন সাত নম্বর দিনের ৮.৪ নম্বর সেকশনের আলোচনা থেকে সেই '/etc/skel' ডিরেক্টরির কথা যার স্কেলিটনের বা কঙ্কালের আদলে একজন নতুন ব্যবহারকারীর নতুন হোম ডিরেক্টরি তৈরি করে নেওয়া হয়? এই '/home' ডিরেক্টরিটা, ব্যবহারকারীদের কল্যাণে, বড্ড দ্রুত ভরে যেতে শুরু করে। অনেকসময় তাই এই ডিরেক্টরিটাকে আলাদা একটা পার্টিশনে দিয়ে দেওয়া ভালো, যাতে বেড়ে গিয়ে গিয়ে মূল রুট পার্টিশনের ফাইলগুলোরই দমবন্ধ করে না-দেয়।

/lost+found — দেখুন তো, এই মহাপ্রভুর নামটা আপনার চেনা চেনা লাগছে, কোথায় যেন শুনেছিলেন? ইএক্সটিটু এবং ইএক্সটিথ্রি ফাইলসিস্টেমের ক্র্যাশ-রিকভারি সংক্রান্ত আলোচনা মনে পড়ছে? আমি আগে এটার কথা জানতাম না। তখনো অব্দি আমি ম্যানড্রেক ব্যবহার করছিলাম, একদিন শখ হল, একটু রেডহ্যাট ইনস্টল করে দেখি। প্রথমত, ইনস্টল করতে করতে এবং বুট করতে করতে চোখে প্রায় জল এসে গেছে, তার আগে অব্দি ম্যানড্রেকে এক্সএফএসের ওই দুর্ধর্ষ গতিতে কাজ করে এসেছি, মনে হচ্ছে সবকিছু যেন স্লো-মোশনে ঘটছে, এরপর ইনস্টল হওয়া সিস্টেমে লগ-ইন করে '`ls`' মেরে দেখি, '/lost+found' জাতীয় মেলো-রোমান্টিক নাম, সুখেন দাস কোথায় লাগে? না, আসলে ব্যাপারটা ততটা মিsটি নয়, ক্র্যাশ-রিকভারির সময় ফাইলসিস্টেম চেক করে পুনরাহরিত ঘেঁটে যাওয়া তথ্য রাখা থাকে এই ডিরেক্টরিতে, ইএক্সটিটু ফাইলসিস্টেমে। অচিরেই রাইজার ফাইলসিস্টেমে রেডহ্যাট ইনস্টল করে আমি মুক্তি পেয়েছিলাম এর থেকে, তার কিছুদিন বাদে রেডহ্যাটটাই উড়িয়ে দিয়েছিলাম। আপনি যদি স্লোমোশন ভালোবাসেন তাহলে ইএক্সটিতে রেডহ্যাট করুন, এবং এই ডিরেক্টরির শোভা দেখুন বসে বসে, কারণ অনেকটা সময়ই তো আপনাকে বসে থাকতে হবে, প্রতিবার কমান্ড দিয়ে প্রম্পট ফেরত আসার জন্যে।

/proc — এই ডিরেক্টরির গোটা ফাইলসিস্টেমটাই অন্য প্রতিটি ডিরেক্টরির থেকে একদম আলাদা এই অর্থে যে এটা কোনো বাস্তব ফাইলব্যবস্থাই না, একটা বিমূর্ত এবং সাঙ্কেতিক ফাইলব্যবস্থা। সিস্টেমের প্রসেসগুলোর বিষয়ে তথ্য যোগানোর উদ্দেশ্যে বানানো একটা নকল ফাইলব্যবস্থা। আমরা, কী কী ফাইলব্যবস্থা আমাদের সিস্টেমে ব্যবহার হচ্ছে সেটা জানার জন্যে, '`cat /proc/filesystems`' কমান্ড ব্যবহার করেছিলাম, মনে আছে? কোনো বাস্তব ফাইল '/proc' ডিরেক্টরিতে থাকেনা। থাকে শুধু সিস্টেম কাজ করার সময়ে সিস্টেমের মেমরি, সিস্টেমে কী কী ডিভাইস মাউন্টেড আছে, সিস্টেমের হার্ডওয়ার কনফিগারেশন ইত্যাদি সিস্টেম-তথ্য। '/proc' ডিরেক্টরিতে গিয়ে, এই গোটা পাঠমালা জুড়ে বাস্তব ফাইল নিয়ে আপনি যা যা জেনেছেন সেটা দিয়ে কদাচ বোঝার চেষ্টা করবেন না। যেমন, আপনি জানেন একটা ফাইলে টেক্সট থাকার সঙ্গে ফাইলের বাইটমাপের সম্পর্ক। এবার '`cat /proc/filesystems`' করে স্ক্রিনে চিহ্নের সংখ্যা গুনে নিয়ে আপনি যদি সেটা দিয়ে '/proc/filesystems' ফাইলটার সাইজের সঙ্গে মেলাতে যান, প্রচণ্ড চমকাবেন। আপনি '`ls -sh`' করে দেখুন গোটা '/proc' ডিরেক্টরির যাবতীয় ফাইলেরই সাইজ দেখবেন শূন্য, 'kcore', 'mtrr', আর 'self' বাদ দিয়ে। এবং মজার কথা কী বলুন তো, আপনার অত্যাবশ্যক পার্টিশনগুলো ছাড়া আর কিছু যখন মাউন্ট করা নেই, মানে আমাদের মাউন্টের ছবিতে স্টপককহীন ওই তিনটে চৌবাচ্চা বাদ দিয়ে আর কাউকেই যখন সিস্টেম 'তুমি যে আমার' করেনি, তখন আপনি '`du -chs /proc`' করে যে সাইজ পাবেন সেটা নাটকীয়ভাবে বদলে যাবে, যেই আপনি পার্টিশনগুলো মাউন্ট করে নেবেন। আমার মেশিনের সুজে সিস্টেমে যখন স্ল্যাকওয়ার, আর্কাইভ আর উইনডোজ পার্টিশনগুলো মাউন্ট করা নেই তখন '/proc' ডিরেক্টরির 'du' মানে ডিস্ক-ইউসেজ বা ডিস্ক-ব্যবহার দেখাচ্ছে চুরানব্বই কিলোবাইট। আর ওগুলো মাউন্ট করা মাত্রই ডিস্ক-ব্যবহার বেড়ে গিয়ে হচ্ছে দুশোপঁচিশ মেগাবাইট। এবং তখনও যাবতীয় ফাইলের আকার শূন্য, একা 'kcore' ফাইলটাই প্রায় ওই গোটা সাইজটা নিয়ে নিচ্ছে। কিন্তু আর না, আমরা আমাদের পাঠমালার এক্তিয়ার ছাড়িয়ে বেরিয়ে যাচ্ছি।

/root — এটা হল রুট ইউজারের বা সিস্টেম অ্যাডমিনিসট্রেটরের নিজের হোম ডিরেক্টরি। '/' বা রুট ডিরেক্টরির মধ্যে এই রুট বা '/root' নামের ডিরেক্টরিতে গুলিয়ে ফেলবেন না। অনেক আগে ওই '/' ডিরেক্টরিটাই ছিল রুট ব্যবহারকারীর হোম-ডিরেক্টরি। সেখানেই থাকত তার নিজের ফাইলপত্তর। পরে, ঘরকন্নার কাজ একটু পরিপাটী রাখার উদ্দেশ্যে এই '/root' ডিরেক্টরিটা বানানো হয়। এই ডিরেক্টরিটা যেহেতু একজন ব্যবহারকারীর হোম ডিরেক্টরি, একে '/home' ডিরেক্টরির মধ্যেই আর একটা সাবডিরেক্টরি, '/home/root' নামেই তো রাখা যেত?

সেটা করা হয়না, তার কারণ, আমাদের মত পিসি ব্যবহারকারীদের জন্যেই শুধু গ্নু-লিনাক্স তৈরি হয়নি। নেটাবদ্ধ বড় বড় সিস্টেমে অনেক সময় ভৌগোলিক দূরত্বে থাকা একটা পার্টিশনকেও মাউন্ট করা হয় '/home' ডিরেক্টরিতে। সেটাও তখন স্টপকক লাগানো চৌবাচ্চা হয়ে যায়। কিন্তু অন্য ব্যবহারকারীদের সাময়িকভাবে বাইরে রেখেও একটা সিস্টেম চালু হলেও, সুপারইউজারকে তো মূল সিস্টেমের অত্যাবশ্যকীয় অংশের মধ্যে থাকতেই হবে। তাই এই ব্যবস্থা। এই সবিশেষ ব্যবহারকারীর একটা আলাদা সুপারভাইজর কোয়ার্টার একদম রুট পার্টিশনের মধ্যেই।

গ্নু-লিনাক্স ফাইলসিস্টেম হায়েরার্কির প্রাথমিক ধারণাটা হল আমাদের। রুট ডিরেক্টরির মধ্যে কী কী ডিরেক্টরি থাকে সেটা আমরা জানলাম। এবার এই ডিরেক্টরিগুলোর চারটেকে নিয়ে আরো দু-চারটে কথা বলার আছে। আলোচনার সূত্রেই বলেছিলাম আমরা, '/bin', '/etc', '/sbin' এবং '/usr' — এই চারটে ডিরেক্টরি আলাদা করে আরো একটু বলে নিতে হবে। সেটা দিয়ে আমরা বরং সামনের দিনের আলোচনা শুরু করব। সেটাই আমাদের এই জিএলটি ইশকুল পাঠমালার শেষ দিন।

আজ চোদ্দই জানুয়ারি। একটু বেশি সময় লাগল এটা শেষ হতে। তবে, গত দুদিন গায়ে জ্বর থাকায় বসে থেকেছি হয়ত, কিন্তু লিখতে তেমন পারিইনি। আর কলেজে পরীক্ষা চলছে, তাই যেতেই হচ্ছে। আপনাদের পড়তে পড়তে কতটা হচ্ছে জানিনা, আমার লিখতে লিখতে ইতিমধ্যেই ক্লান্ত লাগছে। এখনো পর্যন্ত শব্দসংখ্যা একলাখ সতেরো হাজার। আর পুরোনো ফাইলগুলো থেকে দেখছিলাম, জিএলটি পাঠমালার এই ফাইনাল ভার্শনটায় হাত দিয়েছি পয়লা নভেম্বরের আগে না, তার মানে, পঁচাত্তর দিনে একলাখ সতেরো হাজার শব্দ লিখেছি, গড়ে প্রত্যেকদিন ১৫৬০-খানা করে শব্দ লিখেছি, এর মধ্যে ব্রেক গেছে একদিন দুপুর, সাত ঘন্টা, আর একদিন রাত্তিরে তিন ঘন্টা। এছাড়া কলেজ যাওয়ার সময়টুকু বাদ গেছে। সে তুলনায় হয়ত কম লাগছে, কিন্তু তার সঙ্গে যোগ করুন তো মোট ওই অতগুলো ছবি। আর এই লিখতে লিখতেও কী পরিমাণ যে আমায় পড়তে হয়েছে সে আর বলার নয়। সত্যিই বড্ড খাটনি। এটা শেষ হলে গ্নু-লিনাক্স কমিউনিটির কাছে আমার ঋণশোধ, পেয়িং ব্যাক টু দি কমিউনিটি, বোধহয় বেশ ভালোভাবেই হবে। আর মাত্র এক দিন। লে বংকা।

সংকলন ও রচনা : মধ্যমগ্রাম জিএলটি-র (glt-mad@ilug-cal.org) তরফে : ত্রিদিব সেনগুপ্ত